Supporting the analysis of a large coin hoard with AI-based methods

Chrisowalandis Deligio*[1], Karsten Tolle[1], David Wigg-Wolf[2]

[1] Goethe Universität Frankfurt am Main - Germany
[2] Römisch-Germanische Kommission – Germany

*Corresponding author
Correspondence: Deligio@em.uni-frankfurt.de

Abstract

In the project "Classifications and Representations for Networks: From types and characteristics to linked open data for Celtic coinages" (ClaReNet) we had access to image data for one of the largest Celtic coin hoards ever found: Le Câtillon II with nearly 70,000 coins. Our aim was not to develop new processes, but rather to demonstrate how existing tools can be used to support the numismatic task of processing and analysing large complexes of coins, thus validating the enormous potential of IT-based methods. The main steps involved are the pre-sorting of coins by size (denomination), the attribution of individual coins to classes or types, and finally the identification of which coins were struck by individual dies.

The process from digitisation of a hoard as images to an actual die study is lengthy and work-intensive. In testing methods to support each of the steps, we focussed particularly on methods that do not need any prior knowledge of the material, in order to explore whether these methods can be applied to a dataset for which there is no more information than the images themselves. The different steps were evaluated against information provided by the numismatist working on the hoard (class and die attributions of the coins), who was also involved in different stages of the process.

The result is a workflow that can be used in future work on large coin finds, thus supporting numismatists and significantly speeding up the work of identification and analysis.

This paper also presents tools, visualisation methods and extensions that proved useful, both for the individual processes, as well as for communicating with the numismatists and integrating their expertise. Earlier phases of our work were presented at CAA 2022 in Oxford.

Keywords: machine learning, celtic coins, unsupervised learning, classification, object detection

Introduction

In our project Classifications and Representations for Networks (ClaReNet), we are exploring and evaluating the application of computer-based methods for the classification of three different Celtic coin series. One of the series consists of the billon staters attributed to the Coriosolitae, a tribe from Britanny, that were included in the hoard of Le Câtillon II found in Jersey in 2012. We are grateful to our collaboration partner Philip de Jersey and Jersey Heritage for allowing us to work on this huge dataset of 120,000 images (some 60,000 for each side of the coins). Before our work began, Jersey Heritage (including 25 volunteers) had invested a huge amount of labour and time in dismantling the hoard, taking the photos and making a first identification of each coin. They provided us with this data during the course of the project.

In order to analyse such a huge dataset and to generate the data, numismatists conduct a series of tasks with different goals. The process is often a top-down approach, starting with the separation into immediately recognisable groups – in this case different sizes of coins (denominations) –, followed by attribution to individual classes or types of coins, and finally even identifying coins produced by the same die. At the same time, damaged, or poorly preserved or corroded coins may be separated, as they are harder to identify and may require more intensive work. As the granularity increases, so do the skill and domain knowledge required. It is this process of analysing a dataset in the several stages into which a numismatist divides the work that forms the focus of this paper. By specifically using different IT-based methods for each stage, the aim is to use well-tried methods to support the process in terms of time benefits, while at the same time ensuring data quality.

The stages are:

1. Pre-sorting - removing those coins that are fragmented or in bad condition and would therefore interfere with or bias later steps.
2. Sorting by size (denomination) - in our specific case separating the staters from the smaller quarter stater and *petits billons* also contained in the hoard.
3. Identifying the coins by class or type - in this case attributing the staters to the six classes of the standard numismatic classification of the coin series.
4. Identifying the coins struck by individual dies within the six classes - such die studies are a basic numismatic tool for reconstructing production processes and economic networks.

For evaluation of our results we had additional information provided by the Jersey team: a) a list containing the class assignments of the staters as determined by the numismatists, (the ground truth) and b) the results of a die study of the smallest of the six classes of the staters (class VI), containing about 1300 coins. We also included the numismatic expert in all stages of our process, in particular in order to improve the data quality of the ground truth and to receive feedback on the value of the tools and processes that we developed. Finally, we developed a method of visualising the results of the tool for supporting a die study that we developed so that the numismatic expert could test it for the other five classes of staters. It should be noted that the expert estimated the time needed by a single person to fulfil this task in the traditional way, without our tool, would be 150 years.

The paper first introduces the dataset and how it was provided, followed by the pipeline that we developed to mirror the different stages into which numismatists divide such work. First, an object detection method was used to prepare the data and crop the images of the coins, a process which includes size approximation. This was followed by the use of an unsupervised method to filter out corroded or damaged coins and to pre-sort the rest, ideally into classes. Using the result and the information provided, we then moved to a supervised approach on coins that were not filtered or well sorted. A short appendix the visualisation, augmentation and explainability methods that we used to communicate with, involve and implement feedback from the numismatist with whom we were cooperating. The last part focuses on the die study, including methods that were developed to support it, as well as an interactive visualisation tool, powered by Orange.

A look at the bigger picture

Coins as a more-or less standardised mass product have already been analysed and classified with IT tools for many years. For modern coins, with their maschine-based production, many systems are able to reach over 99% accuracy in identification (Modi and Bawa 2012). Ancient coins are very different, since dies for their production were hand made and the coins themselves hand-struck, so that both can vary greatly. On top of this, the quality of preservation of individual coins can differ a lot depending on their intensity of circulation (wear) and how corroded they are. In most cases, the coins used for computer vision approaches have come from collections with many well preserved individual specimens. A good example of this is the Roman Republic Coin Dataset (RRCD) that can be downloaded via Github[1] consisting of some 17546 images of coins of 100 different classes (coin types). While such a dataset is ideal for comparing computer vision approaches, as done by (Anwar et. al. 2021), it does not reflect the reality faced by numismatists who deal with coin finds, as exemplified by the coins and photos dealt with here.

For die studies Heinecke et al. (2021) attempted an automatic approach based on well preserved gold and silver coins, in contrast to our material. So far, attempts at providing an online service or the relevant code have not been successful, although the code for the CADS system (Taylor 2020) is available in GitHub. Essentially, we employed a very similar approach to CADS, but for visualisation of and interaction with the results we decided to simply adopt Orange Data Mining widgets instead of implementing a new system, thus greatly reducing any maintenance problems. We also tailored the system to assist the numismatist in carrying out an actual die study. Generally, it is important to keep the domain expert in the loop, as this

[1] https://github.com/saeed-anwar/CoinNet?tab=readme-ov-file

will help increase the usability and practicability of the pipeline and tools developed, as well as create trust in and acceptance of such approaches by future users.

Data

The data set made available to us consisted of about 60,000 photos for each face of the coins and included not just the staters that we planned to study, but also smaller coins such as quarter staters and *petits billions*. The photos themselves mostly included not only the coins, but also a scale bar, and sometimes further information such as the inventory number assigned to the coin(ID), or occasionally the class to which the numismatist assigned the coin. Both the coins and the photos were of very mixed quality: there were broken, worn and corroded coins, as well as blurred, overexposed and underexposed photos (fig. 1). The variance in the photos was due to them being taken by a number of volunteers without any standardisation. For our projects we focused on the staters, as they are the most common coins in the hoard (about 50,000) and numismatic research on them is more advanced than for the quarters and the *petits billons*. The staters are generally divided by numismatists into six classes (I to VI, Appendix 1), a classification which was originally based on the obverse, or 'heads' face of the coin (Colbert de Beaulieu 1957). Therefore we concentrated our work on the obverse. Central information that we had in order to evaluate our work was:

- Staters have an average diameter of about 22 mm, quarter staters and *petits billions* about 13 mm.
- The numismatist provided us with information about the denomination (size) of the coins (stater, quarter stater, …) and the class they were assigned to (ground truth).
- For one class of the staters (VI for which we had some 1300 images) the numismatist had already carried out an unpublished die study.

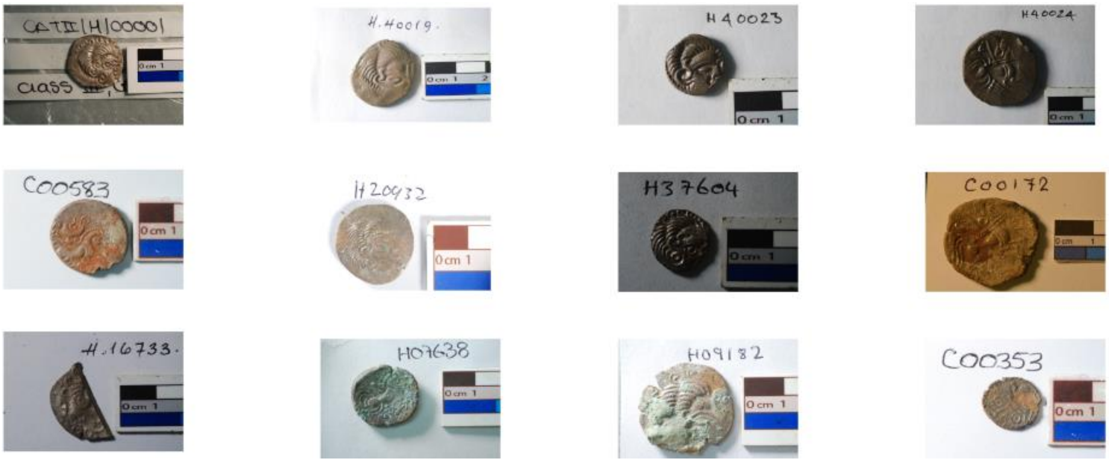


Figure 1 - Variations in the photos and conditions of the coins (Photos: Jersey Heritage).

Overview of the pipeline

In order to identify the staters among the 60,000 coins for which we had images we followed a divide and conquer methodology. The aim is to sequentially divide the dataset into smaller batches in order to analyse each one more efficiently. Our pipeline thus included the following steps:

1. Pre-sorting and size estimation based on **Object Detection** - Since the focus of our work was the image of the coin itself, the coins need to be detected and cropped from the photos. At the same time the size of the coin can be calculated by detecting the scale bar on the photo, thus allowing a first sorting process to identify the staters on the basis of their being larger than the quarters and *petits billon*s.

2. Further pre-sorting based on **Unsupervised learning** - The intention was to use only the images as input, so thatinitially we employed methods that do not require any further domain knowledge. This step of the pipeline was repeated in order to identify groups of high similarity that corresponded to the expert's classification, while removing corroded and worn coins to eliminate any bias they might cause.

3. Classifying the coins by class based on **Supervised learning** - The results of step two were checked against the classification of the expert, and the groups that corresponded to the expert's classification used to train a model to assign the coins from the other batches to the numismatic classes. The domain expert was also involved in the process.

4. Implementing a **die study** - By checking the results against the die study already carried out by the numismatists for one of the classes, we compared different approaches (unsupervised, supervised and feature detection) for their effectiveness in the task. Finally, we implemented a system based on Orange Data Mining in order to support the expert in their ongoing work on the remaining five classes to accelerate the process.

For the steps one to three we used a divide and conquer approach, meaning that we divided the dataset step by step in order to facilitate better analysis at the next step. The first division into large/small and broken coins (second row) is the result of object detection and the calculation of the approximate size of the coins. The second division into high and low quality is produced by the unsupervised method, the last one is the result of the application of size identification (step one) together with step two (high/low quality).

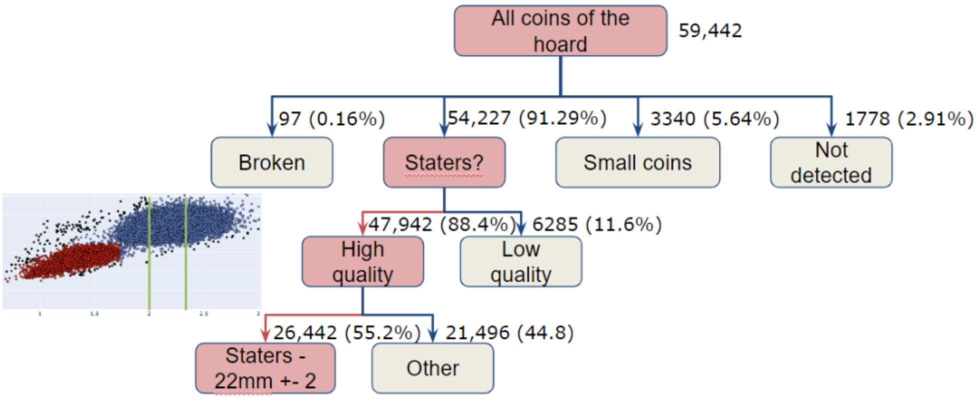


Figure 2 - Using the divide and conquer methodology, the data set could be divided step by step into more easily analysable parts (Graphic: C. Deligio, Big Data Lab).

Object Detection

This step can be seen as a pre-processing step, based on standard approaches in the IT field, with no major training effort needed. Its primary purpose was to crop the coins from the photos so that we could focus on them. At the same time, we were able to use this step to determine the size of the coins using the information in the image itself, thus exploring the potential utility of the method to identify the staters autonomously. As already mentioned, the quality of the photos varied greatly, but their format was generally consistent. For most of the photos, four areas could be defined: the inventory number (ID), the coin, the scale and the class (rare), of which the last two were sometimes missing, in particular the class. For us, the coin and the scale were relevant, while the ID can be retrieved via the file name and the class was available in the spreadsheet provided by the expert (fig. 3).

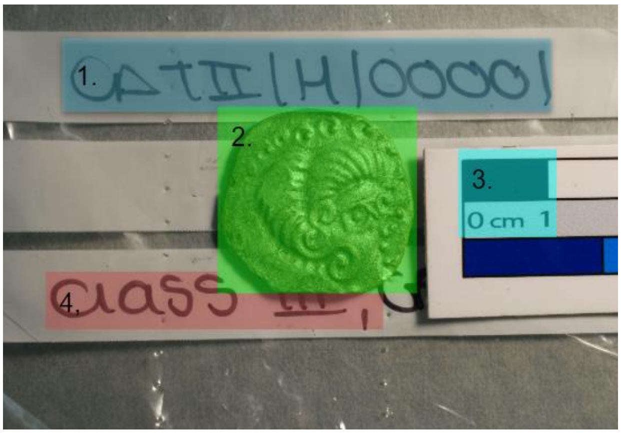


Figure 3 - The positions of the four defined areas are consistent overall. (Photo: Jersey Heritage)

The implementation was carried out with a typical supervised object detection process in which two classes, coin and scale, were defined, and a training and a test dataset were created. As “scale” we defined and labelled the black section of the scale bar representing one centimetre, and not the scale bar itself. It provides the information needed to calculate the size of the coin. The position of the two objects on the images was consistent and the scale was a less complex target due to its representation as a simple black box. Given this consistency, we decided to select only a relatively small data set in order to first evaluate the procedure: 100 images were chosen as training data, and 25 as test data. They were chosen on a more or less random basis, although it was also important to include small and large, as well as broken coins. Annotation of the data was done using the open source tool labelImg (Tzutalin 2015). The evaluation of the test dataset after training gave a mean average precision of 95% for the calculation of the size. The evaluation for the whole dataset could not be given as a percentage, but by calculating the size of the coins it was possible to identify outliers and thus improve the procedure in a targeted way. In addition to the outliers detected by size, we also manually re-measured 10 coins to check the quality of the result, which was indeed accurate. Finally, we cropped the images of the coins and verified whether they were still completely displayed. Although 100 coins was only a small sample, it did prove that this number of images was sufficient for the task. An overall evaluation would require the annotation of all data, which would be extremely time-consuming and is not essential for the further process. For the implementation, we used Tensorflow's Object Detection API[2] and its Model Zoo[3] in order to select a model architecture. We did not attempt to evaluate the different architectures against each other but decided to use the CenterNet Hourglass104 512x512 architecture by Duan et al. (2019) because it had a good balance between time and accuracy. In Model Zoo benchmarks are given that have been calculated on the COCO dataset[4], such as the mean average precision, and these can be used as a guide when choosing an architecture.

Once the model had been trained and tested, it was applied to the entire dataset. Each image was cropped and the size was calculated. An optimal result is shown in Figure 4. The size calculation is intended to sort the data set, rather than to calculate minimum and maximum diameter exactly. Therefore, we

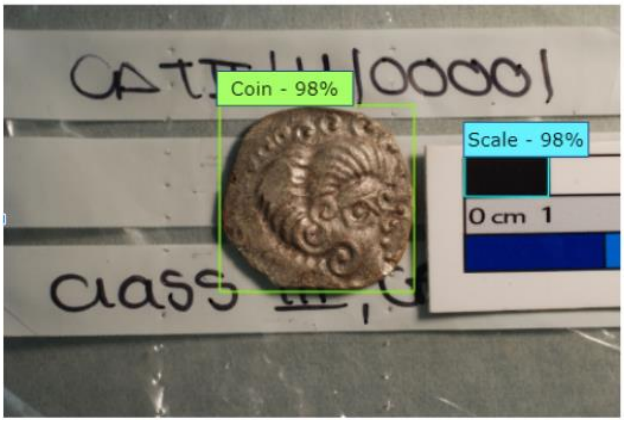


Figure 4 - Optimal Prediction of the model. Calculated values: height: 2.321cm, width: 2.194cm. (Photo: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

[2] https://github.com/tensorflow/models/tree/master/research/object_detection

[3] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

[4] https://cocodataset.org/#home

decided to take only two measurements: the width and height of the bounding box of the coin that was detected. The size of the coin in centimetres is given by dividing these two values for the coin by the width of the bounding box for the scale. The cases where the size calculated for the coin differed greatly from the majority, and was therefore to be classified as an outlier, were considered separately. Figure 5 shows such a case. There are several reasons for an incorrect calculation. Sometimes darker or shadowy areas in the image are erroneously classified as the scale, a problem when the target is a black box. In the example in figure 5 there is also an area for which the same percentage as the scale itself was attributed, causing selection problems during the calculation. Where problems were apparent, the images were annotated, thus broadening the training base, as were other suitable images (e.g. images with shadows).

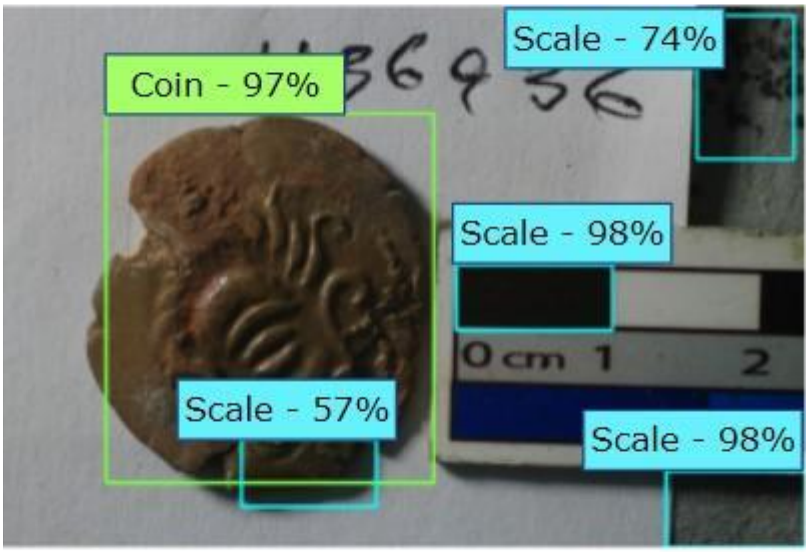


Figure 5 - Shadowy areas can lead to an incorrect prediction by the model, resulting in a false calculation of size. (Photo: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The calculation was made for each photo, i.e. for both sides of the coin, resulting in four measurements for each coin. Comparing the values for both sides can also act as a quality check, and cases where the values were very different were examined more closely. The calculated values in centimetres are visualised in a scatter plot in Figure 6. The colour coding is as follows: coins with a height and width deviation of more than 40% are defined as damaged and marked in black. Small coins (probably quarter staters and petit billions) are marked in red and large coins (probably staters) are marked in blue. Where we can separate the different groups exactly needs to be examined more closely, but the visualisation shows that there is a gap at about 1.75cm, so we chose this as the boundary. The blue area is the focus of our work because it should contain the staters. But as we want to analyse it in more detail first, we will call this group the 'Staters?'.

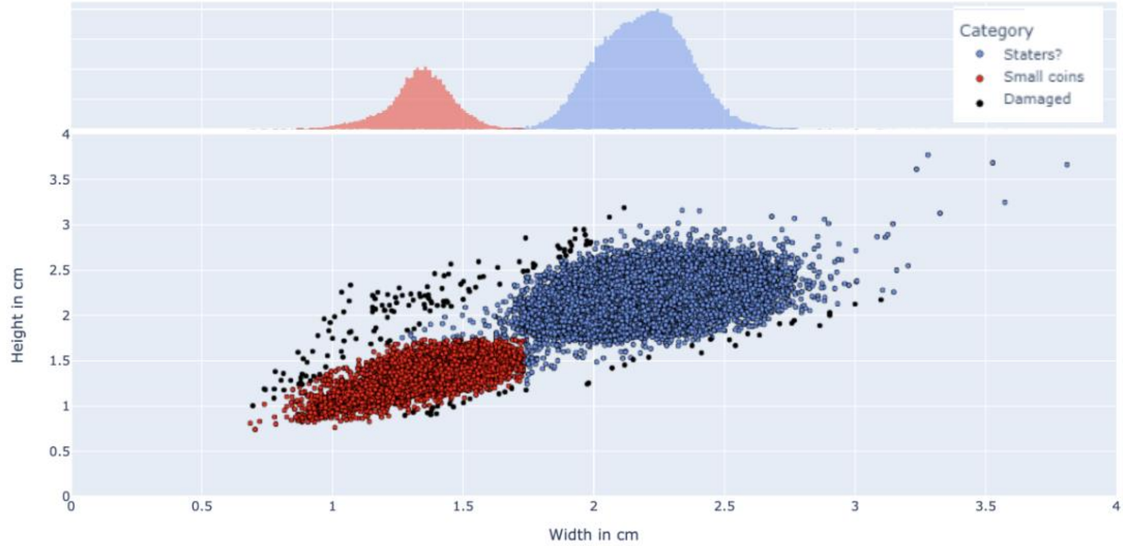


Figure 6 - Scatter plot of the approximated diameter. (Graphic: C. Deligio, Big Data Lab)

If we look at the peaks of the two point clouds, most of the results are at around 1.3 cm and 2.2 cm. These two values correspond to the information provided by the expert and thus provide first confirmation of the validity of our process. The dataset can therefore be divided into four groups: "Staters?" (54.227 - 91.29%), "Small Coins" (3.340, 5.64%), "Damaged" (97, 0.16%) and "Not Detected" (1.778, 2.91%) (figure 2). The latter contains photos where no scale was available or it was not detected, and therefore no calculation could be carried out.

Unsupervised Learning for pre-sorting

Taking our divide and conquer approach one step further, we selected the "Staters?" group as the dataset for unsupervised learning. The main aim was to see if we can pre-sort the coins into the numismatic classes successfully and compare the results with the classification by the numismatist. Deep Learning and convolutional neural networks (CNNs) were our first choice for image classification, and a promising approach proved to be the DeepCluster (Figure 7) method developed by Caron et al. (2018). This combines a convolutional neural network with a clustering algorithm for unsupervised training of a CNN. The idea behind this approach is to use the clusters generated as pseudo-labels to train the CNN, and the extracted features in turn serve as input to the clustering algorithm. This process is then repeated for the desired number of epochs.

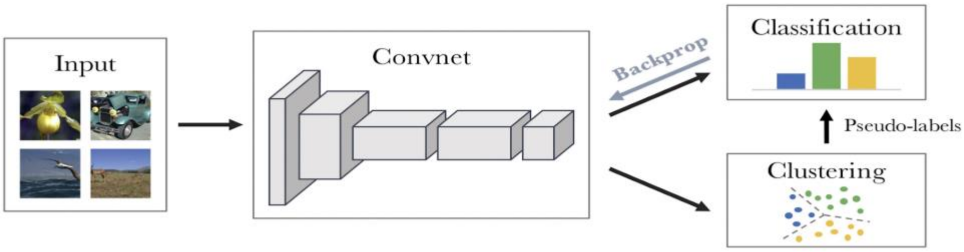


Figure 7 - DeepCluster implements a method of unsupervised training of a CNN (Caron et al. 2018).

We used the VGG16 architecture, and as the clustering algorithm k-Means. The required inputs for the system were the images of the coins and the number of desired clusters (k). The choice of k was initially a challenge, for Caron et al. recommend a much larger k (e.g. the best results are obtained with a k 10 times larger than the actual number of classes). On the other hand, the number of expected classes was known (6), but since we wanted to analyse the dataset without using any prior information, we started with a k equal to 100. To measure another factor of the effectiveness of the method, for our first exercise we entered both the obverse and reverse photos of the coins to see if they would be separated. For the evaluation of the resulting clusters, we avoided the use of any additional information and performed it manually. Based on this manual evaluation, the following observations were made (fig. 8):

- Obverse and reverse were generally not mixed within the clusters → showing that the method already works at a high level.
- Coins in poor condition were grouped together → showing the potential to clean up the dataset for further processing.
- Clusters with different levels of wear were identified.
- There were mixed clusters with no common features → CNNs are complex, often being described as black boxes, and it is not always clear how the clusters were generated.

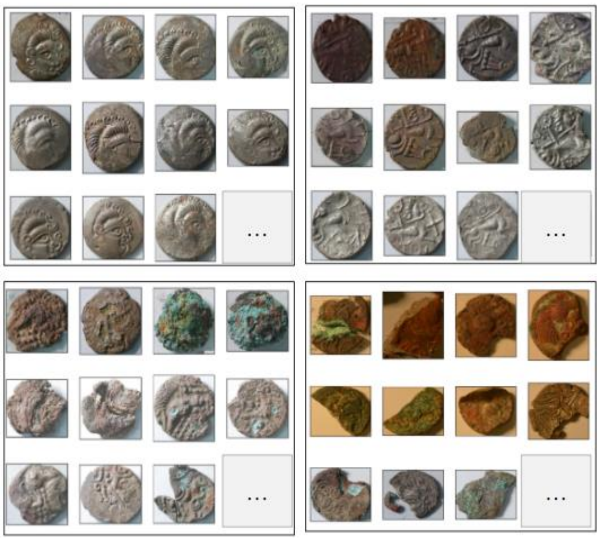


Figure 8 - Example of clusters generated. (Photos: Jersey Heritage)

Many clusters showed strong similarity (based on our manual evaluation), while clusters which had identified corroded and poorly preserved coins were sorted out. This allowed us to divide the dataset into "High Quality" - clusters with a high similarity and well preserved coins - and "Low Quality" - corroded and worn coins - as well as clusters with a significant mix. In order to have a further degree of certainty we focused on the coins at the centre of the size point cloud, i.e. 2.2cm +- 0.2cm (fig. 5). This was done to be absolutely sure that only actual staters were present in the dataset (fig. 8). Additionally, for the next step we only used the obverse images.

The goal for this new, reduced dataset (c. 26,000 images) was to divide them into the six existing classes for staters defined by numismatists. Running the unsupervised approach with just six classes would generate clusters that are too big and not be very useful, so we decided to run it with k=25 (25 clusters), which we then evaluated against the spreadsheet provided by the numismatists of his class attributions (ground truth). The first evaluation was to determine whether our selection did contain only staters, something that was confirmed. By comparing our results with the expert's spreadsheet, we could also check exactly which classes were represented in each cluster: the result was that 18 out of 25 clusters (15,063 images) contained at least 79% of coins of only one single class (the actual values ranged from 79% in cluster 7 to 99,7% in cluster 20). Of those 18 clusters, only 8% of the coins (1208 images) did not correspond to the class given in the numismatist's spreadsheet. The evaluation of the clusters is presented in Appendix 2. One negative result was that we did not manage to find any clusters with only class VI coins, partly due to the fact that it is by far the smallest class in the dataset with only some 1300 examples present in the hoards. The images of class VI coins were mostly mixed into clusters of class V, or sometimes IV, which is probably a result of the relative similarity with these classes. In total, the attribution by the CNN of 13,855 coins (23% of the total data set of c. 60,000 coins) were confirmed by the comparison with the spreadsheet. This data set, verified by two systems, 1) by deep learning and 2) the expert's classification, now formed the basis for the supervised approach (represented by the green segment in the pie chart in fig. 9).

From unsupervised to supervised

This validated dataset was now used as the basis for building a supervised, trained CNN model. As the dataset was highly unbalanced (ranging from 615 to 5317 images per class) and we could not automatically extract class VI with our method, we adapted the dataset slightly. In order to be able to identify class VI, we added the coins validated by the expert as class VI. Furthermore, we rebalanced the dataset by downsampling to the smallest class. For the CNN architecture we again used VGG16. After training, we used this model to predict the coins that had been clustered wrongly (represented by the red segment of the pie chart in fig. 9). For the outcome we involved the numismatic expert because the predictions could have two results: where the prediction matched the ground truth of the spreadsheet, it was basically confirmed. Cases where the prediction differed were not automatically classified as false, but instead saved separately to be reviewed.

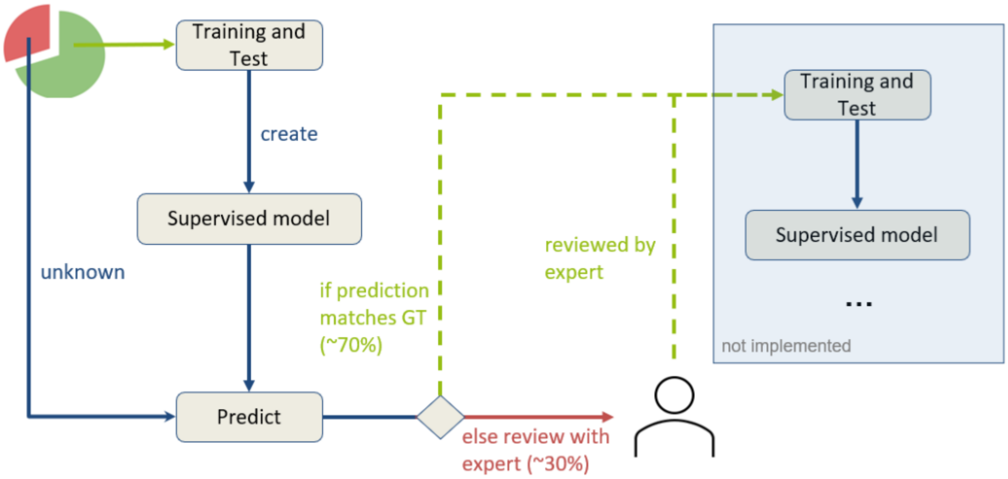
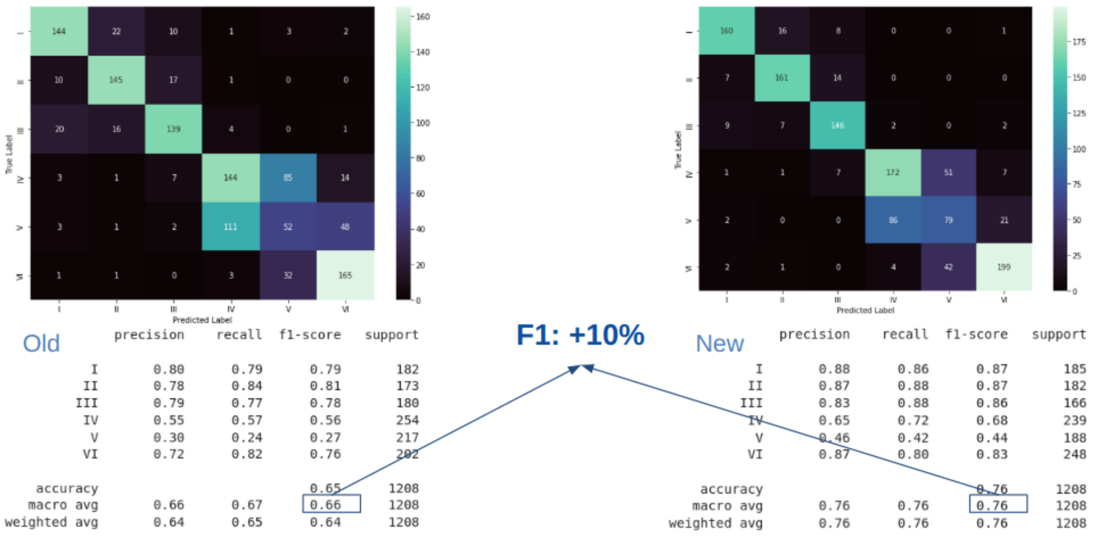


Figure 9 - The 13,855 images from the previous step were selected as the first training base (green). The 1208 wrongly selected images were used as the first test set (red). The percentages are the results of the predictions for the test set. (Graphic: C. Deligio, Big Data Lab)

Whether the prediction of the CNN or the class given in the ground truth provided by the numismatist was correct had to be decided by someone with domain knowledge, and therefore we involved the numismatist. To do this we created a list with the images of the coins with the ID, together with two values: the CNN's prediction and the class assignment in the ground truth. But which was which was masked so that the numismatic expert could not see it. He could then choose which value was correct, or even specify a new one. Comments by the expert were also encouraged, especially in the case of difficult decisions, as they could help us as non-experts to understand the difficulties. In our first test set (illustrated in Figure 9), the CNN model diverged from the ground truth for 30% of the coins (328 out of 1208 images). The review by the expert for these cases could be divided into four cases:

1. The class assigned in the ground truth was actually wrong and was improved by the model --> data quality improvement (115 cases - 35%).
2. The class assignment was not clear --> problematic cases (26 cases - 8%)
3. The model was wrong (175 cases - 53%), mostly between class IV and V (126 cases)
4. Both were wrong (12 cases, c. 4%)

The model mainly had problems distinguishing between classes IV and V, but the expert had the same problem. We also distributed the list to our project team (i.e. to non-experts) and they also had problems with exactly these two classes. Figure 10 presents the CNN's evaluation of the 1208 images with the old classification (the one originally supplied by the expert), together with the result when the expert's revised classification is taken into account. With the revised classification, the F1 metric increased by 10%. This demonstrates above all that the performance of such a model cannot be calculated exclusively on the basis of metrics, but that the underlying data quality must be taken into consideration.



True Label \ Predicted Label	I	II	III	IV	V	VI
I	144	22	10	1	3	2
II	10	145	17	1	0	0
III	20	16	139	4	0	1
IV	3	1	7	144	85	14
V	3	1	2	111	52	48
VI	1	1	0	3	32	165

```
              precision    recall  f1-score   support

           I       0.80      0.79      0.79       182
          II       0.78      0.84      0.81       173
         III       0.79      0.77      0.78       180
          IV       0.55      0.57      0.56       254
           V       0.30      0.24      0.27       217
          VI       0.72      0.82      0.76       202

    accuracy                           0.65      1208
   macro avg       0.66      0.67      0.66      1208
weighted avg       0.64      0.65      0.64      1208
```

True Label \ Predicted Label	I	II	III	IV	V	VI
I	160	16	8	0	0	1
II	7	161	14	0	0	0
III	9	7	146	2	0	2
IV	1	1	7	172	51	7
V	2	0	0	86	79	21
VI	2	1	0	4	42	199

```
              precision    recall  f1-score   support

           I       0.88      0.86      0.87       185
          II       0.87      0.88      0.87       182
         III       0.83      0.88      0.86       166
          IV       0.65      0.72      0.68       239
           V       0.46      0.42      0.44       188
          VI       0.87      0.80      0.83       248

    accuracy                           0.76      1208
   macro avg       0.76      0.76      0.76      1208
weighted avg       0.76      0.76      0.76      1208
```

Figure 10 - Same predictions, different results. Comparison between two classifications (old vs revised). (Graphic: C. Deligio, Big Data Lab)

A next step would be to improve the supervised model by repeating the process (the blue box in Fig. 9) with the remaining coins, starting by predicting the classes of the coins in the remaining seven of the original 25 stater clusters. This process could then be repeated step by step for the remaining coins that had been excluded in the pre-sorting process.

An important question we also asked ourselves was how to deal with the coins that we defined as 'low quality': to what extent could our model be useful. Using a random sample of 20 images per class from this set, we conducted a small case study. It was of particular interest to see if the model that had been trained on very good images could be applied. Our case study achieved an accuracy of 47%. Figure 11 shows the confusion matrix. The figure also shows three examples visualised with GradCam. It can be seen that the regions that are of most relevance to the class assignment (such as the hair or eye) are focussed on, but that there is also a bias due to the condition of the coins as these are their best preserved areas. The two images on the left are correctly classified by the model. Comparing the right-hand image with the images of the individual classes in Appendix 1, it could indeed very well be class III (based on the style of the eye).

This was only a preliminary test, but it demonstrated not only the important role played by the ground truth, but also that the ground truth can and should be questioned. It is also clear that in order to improve the model, more such material should be integrated to counteract the bias of the condition of the coins.

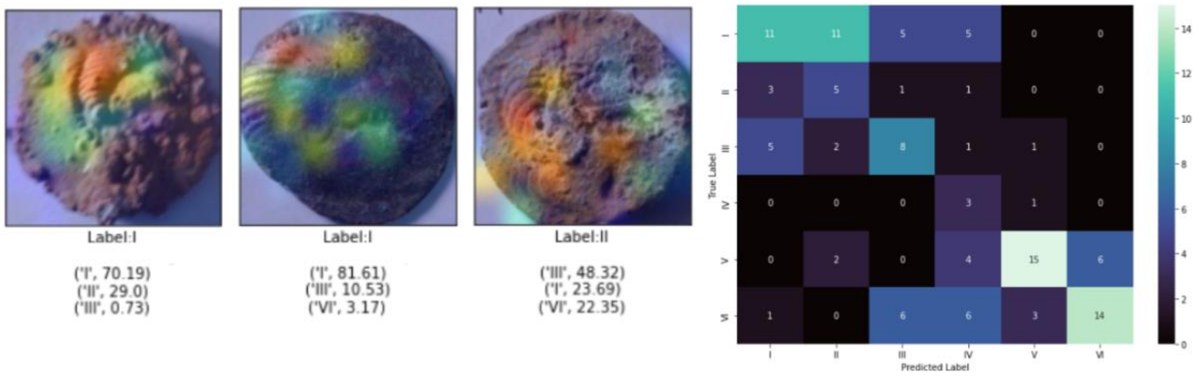


Figure 11 - left: the visualisation of the prediction with the top 3 values; right: the matrix of the 120 predictions. An F1 value of 44% and an accuracy of 47% were achieved. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

Supporting a die study

In order to investigate how we might support a die study, we dug further into a single class in order to try to identify coins struck with individual dies. As noted above, for one of the six classes (VI with about 1300 images) an unpublished die study was available to us in order to evaluate our methods. Die recognition brings new challenges for, although there is less data to deal with, the coins are very similar (being of the same class) and there are many dies. Previously we had about 60,000 images with the goal of distinguishing six classes, but now we had only 1300 images of coins struck with some 30 or more dies (based on the unpublished die study). For the implementation, we tested three methods against each other:

1. Reapplying DeepCluster (k=45, as an approximation of the defined die classes).
2. Using our trained supervised model to extract features and then cluster them.
3. Employing algorithms that compare the key points in the image, which has been successfully used on other coinages.

The first two methods are similar in principle but differ in the trained CNN that is used. In (1) the CNN is trained from scratch using the DeepCluster algorithm, in (2) we used our model that was trained on the six classes in order to extract features. (3) is a very different method to the CNNs and is discussed briefly here. The algorithms used are from the field of image matching, the best known of which are probably SIFT (Scale-invariant feature transform by Lowe, 2004), and SURF (Speeded Up Robust Features by Bay et al., 2006). Both algorithms are patented, but a popular open source alternative is ORB (Oriented FAST and rotated BRIEF by Rublee et al., 2011). Feature matching algorithms have been successfully applied to ancient coins in various publications (Kampel and Zaharieva 2008; Taylor 2020; Heinecke et al. 2021) and for our procedure we used ORB. Some pre-processing steps were employed that had a positive effect on the results in terms of reducing bias (such as scratches arising from the coins' use and wear). The images were converted to greyscale in order to avoid a colour bias and were blurred and contrast adjusted (see Heinecke et al. 2021). Finally, a circle crop was applied to remove the edges of the coins and to focus only on the motif. Examples of the input are shown in figure 12.

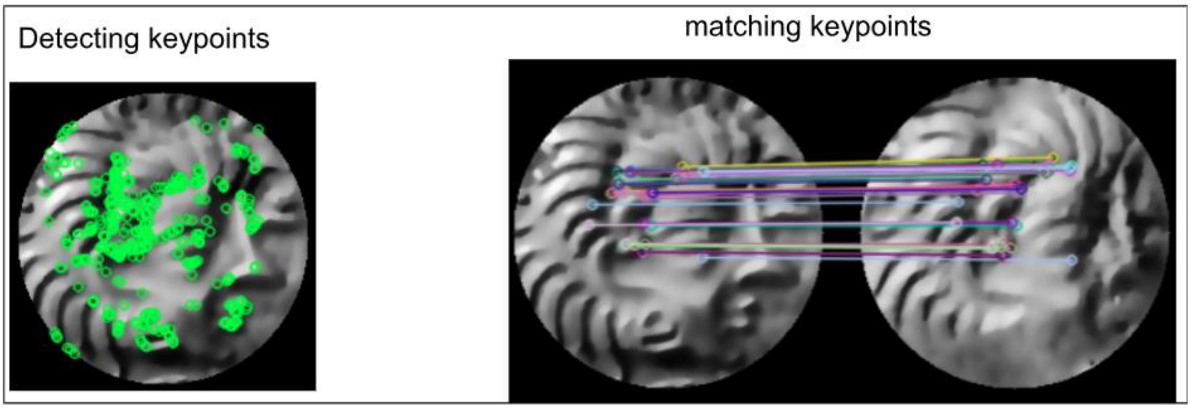


Figure 12 - Left, an image with detected keypoints. Right, an example of two coin struck with the same die according to the ground truth and their matches. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The process is as follows: the first step consists of key point detection and matching the key points between two images. This comparison is carried out in pairs between all images, a process that mirrors how numismatists conduct analogue die studies, comparing each coin individually with all other coins. The matches found are captured as a vector for each image (resulting in an n x n matrix). The exact method of calculating the key points in ORB is explained in Rublee et al. (2011).

The second step is the same for all three methods. The features from (1) and (2), which are also stored as a vector, and the result from (3) are used as input to a clustering algorithm, in our case hierarchical clustering. We used the Orange Data Mining tool for calculating distances (based on the Spearman distance metric), clustering and visualisation. Figure 13 shows the visualisation of the clustering as a dendrogram, using a hierarchical clustering with a complete linkage (already implemented in Orange). The result of the clustering is compared with the existing die study carried out by the expert. In order to evaluate the three methods equally, there are various possibilities; we decided to evaluate them all with the same distance value[5]. We started with the image matching method (3), for which the distance value 0.3 proved to be optimal, and this was also chosen for the other two methods for direct comparison. Table 1 is a summary of the results that were obtained with this value. It shows the best value achieved within a cluster in terms of the number of coins from an individual die that were identified, the mean value of all clusters and the total number of clusters formed.

[5] To calculate the distance we used the Spearman metric.

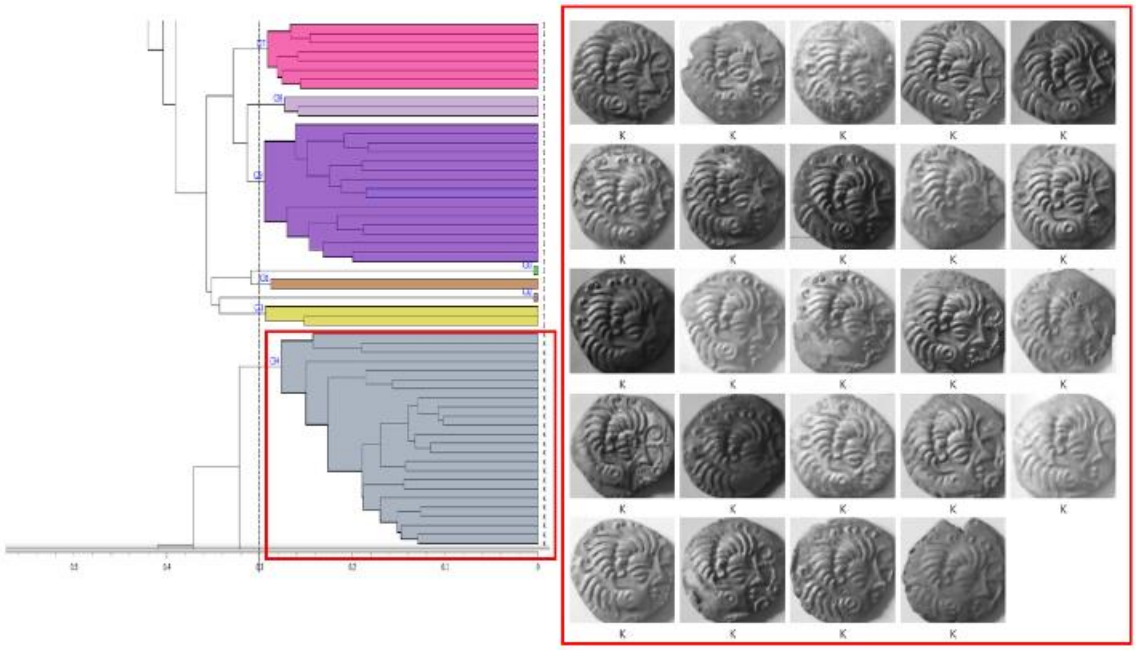


Figure 13 - Left, part of the dendrogram created with the hierarchical clustering widget in Orange using the Spearman distance, based on the results of ORB. Right, overview of a cluster that contained only coins struck by one die. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The DeepCluster method performs better with a larger amount of data and in this case struggled with the relatively high number of classes for the small number of images. The second method, our supervised trained model, performed somewhat better, with the added advantage that the model already existed and no additional training time was required. The third method based on ORB worked best. Looking more closely at the results of the third method, of the 256 clusters, 208 had at least 70% coins from the same die. Furthermore, 194 of them had only coins of just one die. These 194 clusters contained 489 coins, that is 40% of the dataset.

Tabel 1 - Supporting a die study. To compare the methods we used the same threshold (0.3).We calculated two values to evaluate the performance: Highest correspondence with the ground truth in a cluster, and the mean for all clusters.

Method	DeepCluster (k=15)	Supervised model (CNN)	Keypoint detection & matching (with ORB)
Nr. of clusters at distance threshold (0.3)	45	172	256
Highest correspondance	60%	75%	100%
Mean	24%	37%	84%

Recapitulation and outlook

We started this research by treating the dataset as if it were a case study of a new find, with no information about the coins available at the outset. With the first step of object detection it was possible to automatically crop the images and, using the scale bar, to calculate the size of the coins and to carry out pre-sorting, in this way helping identify the staters, which were the coins with which we wished to work. The IT methods used are standard and in our case required little effort/training.

The next step of unsupervised learning still does not need any input from numismatists as domain experts.-but the resulting clusters have to be evaluated manually. A first process with 100 clusters allowed us to exclude about 12% of the coins, which were identified as unsuitable, badly preserved pieces. The data set was further narrowed down by taking only staters with a calculated diameter of 22mm +-2mm (the standard size as defined by the numismatist), and the unsupervised method was repeated to produce 25 clusters. Since we had received a spreadsheet from the numismatic expert providing his classification of the coins, it was possible verify that a) this dataset did indeed contain only staters, and b) that 18 of the 25 clusters mainly contained coins of the same class. The best result was 0.997% (cluster 20: only two of 772 coins are not of the same class). In a situation where the domain expert has not yet classified the coins, it is clear that the presorting into clusters would significantly speed up his task (as is also the case for the method to support a die study that we developed). However, generating a ground truth is mandatory for training a supervised classification model.

Re-evaluating the data with a supervised method led to a significant improvement in data quality. It also showed that experts and AI had similar problems, in particular in distinguishing classes IV and V, which may indicate that the border between the classes is not sharp. Our experience showed that in such cases it is necessary to involve the domain expert in the evaluation process. Specific modifications can be made to influence the areas or features on which the AI concentrates in order to create a model that is closer to the criteria employed by the numismatic expert, for example concentrating on specific areas (e.g. the nose). But since this was not within the scope of our project, we only briefly looked into this direction.

A CNN is a system that learns to distinguish different classes, which includes learning different levels of abstraction from the individual coins. When it comes to studying coins, however, such abstractions are generally not productive because in most cases the amount of data is too small and the differences between the coins can be minimal. It therefore proved useful to take a look at other algorithms (in terms of complexity and computing power required), in our case image matching (ORB). This approach is particularly suitable for small datasets, especially when there are many classes and a high similarity of data, as is shown by the positive results for our die study. Selecting the right approach and algorithm is still a challenge, even for IT experts. This is partly due to the fact that each approach also requires various subtasks (e.g. different methods of preprocessing or augmentation) and has different possibilities for fine tuning (e.g. the number of clusters, hyperparameter settings, choosing the best loss function for the task).

The results of our work clearly demonstrate that semi-automatic processes can be extremely helpful in sorting and classifying large complexes of coins, and can even support a work-intensive and time-consuming die study. We believe that the system we built around Orange Data Mining will speed up the die study for the other five classes of staters in the Me Câtillon hoard. Furthermore, our experience has shown that a human centric approach that involves close cooperation with domain (numismatic) experts can be a good way to increase trust and acceptance of IT methods and achieve a high success rate.

Data, scripts, code, and supplementary information availability

Implementations used in this paper:: https://github.com/Frankfurt-BigDataLab/2023_CAA_ClaReNet

Official implementation of DeepCluster by Caron et al.: https://github.com/facebookresearch/deepcluster

Annotation tool used: https://github.com/heartexlabs/labelImg

Implementing an object detection model: https://github.com/sglvladi/TensorFlowObjectDetectionTutorial

For implementing a supervised model: https://www.tensorflow.org/tutorials/images/transfer_learning

Useful augmentations library: https://albumentations.ai/

Tool for visualising results (and more): https://orangedatamining.com/

Conflict of interest disclosure

The authors declare the following non-financial conflict of interest: David Wigg-Wolf is recommender for PCI Archaeology.

Funding

The Classifications and Representations for Networks (ClaReNet) project is funded by the German Federal Ministry of Education and Research (BMBF).

References

Anwar H, Anwar S, Zambanini S, Porikli F. Deep ancient Roman Republican coin classification via feature fusion and attention. Pattern Recognition. 2021 Jun 1;114:107871

Bay, Herbert, Tinne Tuytelaars, Luc Van Gool. 2006. *SURF: Speeded Up Robust Features*. In: Leonardis, A.,Bischof, H., Pinz, A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11744023_3

Caron, Mathilde, Piotr Bojanowski, Armand Joulin, Matthijs Douze. 2018. *Deep Clustering for Unsupervised Learning of Visual Features.* https://doi.org/10.48550/arXiv.1807.0552

Colbert de Beaulieu, Jean-Baptiste. 1957. *Le trésor de Jersey-11 et la numismatique celtique des deux Bretagnes*. Revue Belge de Numismatique Vol. 103. 47-88.

Duan, Kaiwen, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang and Qi Tian. 2019. *CenterNet: Keypoint Triplets for Object Detection*. https://doi.org/10.48550/arXiv.1611.10012

Heinecke, Andreas, Emanuel Mayer, Abhinav Natarajan, Yoonju Jung, *Unsupervised Statistical Learning for Die Analysis in Ancient Numismatics*. Computer Vision and Pattern Recognition 2021. https://doi.org/10.48550/arXiv.2112.00290

Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama and Kevin Murphy. 2017. *Speed/accuracy trade-offs for modern convolutional object detectors*. https://doi.org/10.48550/arXiv.1611.10012

Kampel, Martin, Maia Zaharieva, Recognizing Ancient Coins Based on Local Features. In: G. Bebis/R. Boyle/B. Parvin (Hrsg.), Advances in Visual Computing. ISVC 2008. Lecture Notes in Computer Science, vol 5358. (Berlin / Heidelberg 2008) 11-22. https://doi.org/10.1007/978-3-540-89639-5_2

Karami, Ebrahim, Siva Prasad, Mohamed Shehata. 2017. *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*. https://doi.org/10.48550/arXiv.1710.02726

Lowe, David G. 2017.*Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision 60, 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Lundberg Scott M., Su-In Lee. 2017. *A Unified Approach to Interpreting Model Predictions*. Advances in Neural Information Processing Systems 30 (NIPS 2017)

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra. 2019. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*. https://doi.org/10.1007/s11263-019-01228-7

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. *Why should i trust you?: Explaining the predictions of any classifier.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige and Gary Bradski. 2011. *ORB: An efficient alternative to SIFT or SURF*. *2011 International Conference on Computer Vision*, Barcelona, Spain, pp. 2564-2571, doi: https://doi.org/10.1109/ICCV.2011.6126544.

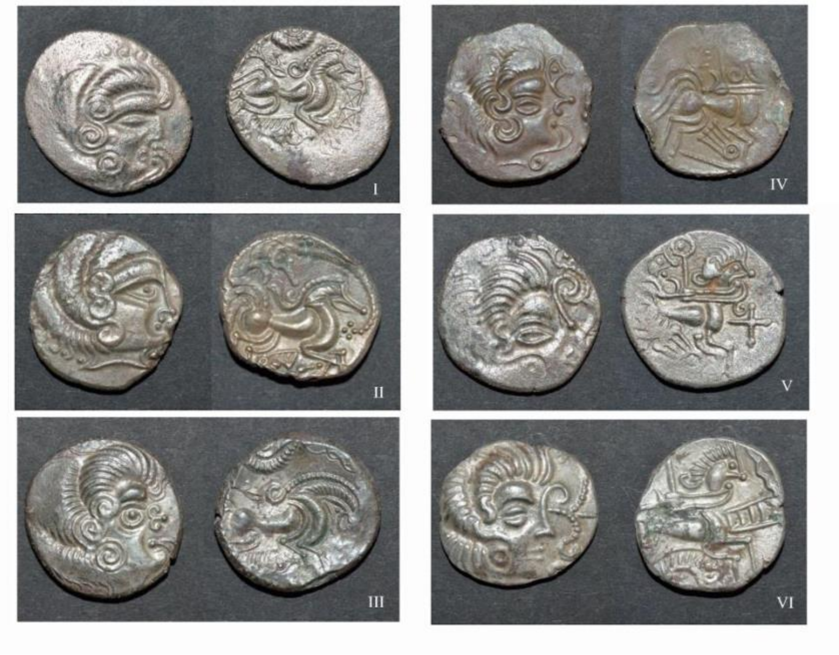
Shatrughan, Modi and Seema Bawa. Image Processing Based Systems and Techniques for the Recognition of Ancient and Modern Coins (2012)

Taylor, Zachary McCord, *The Computer-Aided Die Study (CADS): A Tool for Conducting Numismatic Die Studies with Computer Vision and Hierarchical Clustering* (2020). Computer Science Honors Theses. 54.

Tensorflow Object Detection API. <https://github.com/tensorflow/models/tree/master/research/object_detection#tensorflow-object-detection-api; 11.04.2023>

Tzutalin. LabelImg. Git code (2015). <https://github.com/tzutalin/labelImg; 11.04.2023>

Appendices



Appendix 1 - The six classes of staters as defined by numismatists. (Photos: Jersey Heritage)

Cluster	Class_I	Class_II	Class_III	Class_IV	Class_V	Class_VI	Other	Total	Class_I_%	Class_II_%	Class_III_%	Class_IV_%	Class_V_%	Class_VI_%
0	0	8	649	9	0	0	11	677	0.0	0.012	**0.959**	0.013	0.0	0.0
1	0	21	646	2	0	0	6	675	0.0	0.031	**0.957**	0.003	0.0	0.0
2	1140	24	7	0	7	0	7	1185	**0.962**	0.02	0.006	0.0	0.006	0.0
3	0	0	0	153	966	55	1	1175	0.0	0.0	0.0	0.13	**0.822**	0.047
4	22	1120	24	0	0	0	34	1200	0.018	**0.933**	0.02	0.0	0.0	0.0
5	2	721	7	0	0	0	8	738	0.003	**0.977**	0.009	0.0	0.0	0.0
6	3	1	2	501	802	130	20	1459	0.002	0.001	0.001	0.343	0.55	0.089
7	0	0	0	6	762	197	0	965	0.0	0.0	0.0	0.006	**0.79**	0.204
8	995	1016	660	572	1214	418	366	5241	0.19	0.194	0.126	0.109	0.232	0.08
9	0	7	490	27	0	0	2	526	0.0	0.013	**0.932**	0.051	0.0	0.0
10	13	842	2	0	5	0	18	880	0.015	**0.957**	0.002	0.0	0.006	0.0

Cluster	Class_I	Class_II	Class_III	Class_IV	Class_V	Class_VI	Other	Total	Class_I_%	Class_II_%	Class_III_%	Class_IV_%	Class_V_%	Class_VI_%
11	647	293	349	29	23	5	34	1380	0.469	0.212	0.253	0.021	0.017	0.004
12	0	11	543	12	0	0	3	569	0.0	0.019	**0.954**	0.021	0.0	0.0
13	0	1	0	615	110	4	20	750	0.0	0.001	0.0	**0.82**	0.147	0.005
14	881	5	5	5	3	0	22	921	**0.957**	0.005	0.005	0.005	0.003	0.0
15	0	1	0	40	1250	156	1	1448	0.0	0.001	0.0	0.028	**0.863**	0.108
16	124	89	65	42	213	24	205	762	0.163	0.117	0.085	0.055	0.28	0.031
17	1	733	7	0	6	0	29	776	0.001	**0.945**	0.009	0.0	0.008	0.0
18	72	105	112	24	38	7	25	383	0.188	0.274	0.292	0.063	0.099	0.018
19	1	502	3	0	0	0	1	507	0.002	**0.99**	0.006	0.0	0.0	0.0
20	0	720	1	0	0	0	1	722	0.0	**0.997**	0.001	0.0	0.0	0.0
21	11	13	630	1	1	0	5	661	0.017	0.02	**0.953**	0.002	0.002	0.0
22	248	297	209	15	7	3	14	793	0.313	0.375	0.264	0.019	0.009	0.004
23	396	0	3	620	335	4	3	1361	0.291	0.0	0.002	0.456	0.246	0.003
24	1	681	1	0	0	0	5	688	0.001	**0.99**	0.001	0.0	0.0	0.0

Appendix 2 - The result of the clustering for ~26,000 obverse images with k=25. Values above the threshold of 0.7 are shown in green.

Appendix 3: Visualisations and Augmentations

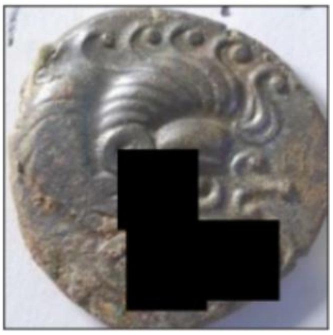


Figure 14 – Cutout random parts of the image. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The involvement of the expert also produced important insights, for example that the form of the nose plays a central role for him when assigning the class. But the visualisations by GradCam of the predictions did not reflect this focus. In the example of the coin in figure 15 (left), the correct class is indeed among the top three values, but with only a 12.8% certainty. The features that are important for the numismatist clearly do not always receive as much weight in the CNN. To address this issue and to try to incorporate the insights of the domain expert, as well as to influence the training process, we tried several augmentation methods. Two of them turned out to be particularly productive. The cutout method involves hiding parts of the image, thus leading the CNN to pay attention to other areas (fig. 14). The cutout can be targeted or randomised, and used with a fixed seed to replicate the training. Looking at the same coin with the model trained with the cutout augmentation (fig. 15b), we can see that certain areas now have stronger weighting, especially the eye region.

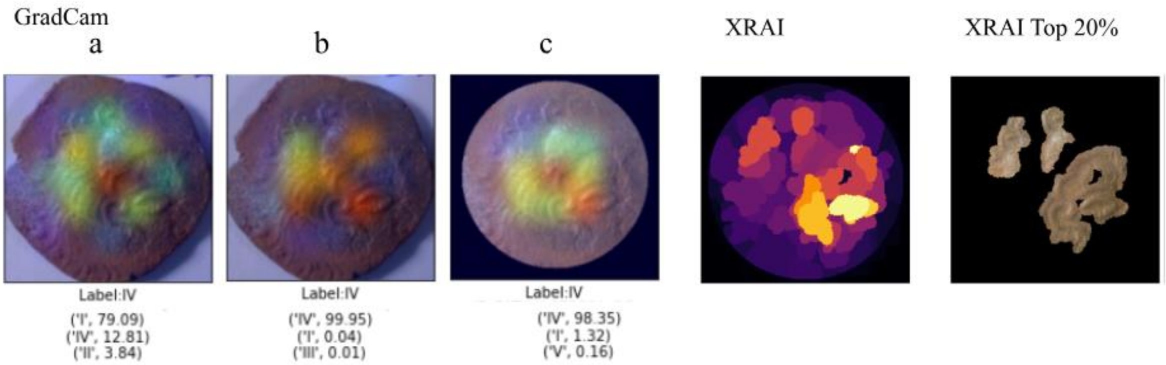


Figure 15 - Left: GradCam visualisation and the top 3 predictions of a model trained based on full coin (a), cutout (b) and circle crop (c) images. Colour scale: blue (weak) - red (strong). Right: visualisation with the use of XRAI (using circle crop images) and cropping the top 20% area. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

Another augmentation, which we call circle crop, was chosen because sometimes the edge of the coin, which can be very irregular, is focussed on by the CNN and so can cause noise. To counteract this, we applied a simple circle crop oriented on the centre of the image to remove the edges. Figure 15 (c) shows that the focus of GradCam was very much in the centre, but less weighted compared to the cutout augmentation. In both cases the class has been correctly identified.

Clearly it is possible to direct the focus of the CNN a little, offering the possibility of incorporating domain knowledge, and to some extent also preferences. When it comes to explaining how a CNN works to non-experts such as numismatists, things can quickly get complicated since their complexity and the large number of parameters in CNNs are difficult to understand. There are various explanatory methods for overcoming this (SHAP, Lundberg and Lee 2017; LIME, Riberio et al. 2016; XRAI, Kapishnikov et al. 2019; and many more), including visual ones such as the GradCam and XRAI methods demonstrated here. We recommend trying different methods and to communicate with the (numismatic) team to find a suitable one. While GradCam expands from one point and is more coherent, XRAI is meant to be independent and more focused on the relevant features regardless of the location (Kapishnikov et al. 2019). The implementation[6] used also offers the option of extracting the most important features (e.g. fig. 15 XRAI Top 20%) instead of displaying a heat map, which was well received by the team and could be interpreted quickly.

[6] https://github.com/PAIR-code/saliency