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 our project "Classifications and Representations for Networks: From types and characteristics to linked open data for Celtic coinages" (ClaReNet) we had image data for one of the largest Celtic coin hoards ever found: Le Câtillon II with nearly 70,000 coins. In the initial stages of our approach, the main problem was how to deal with the dataset without having any information about it. First, we separated the dataset into groups of coins of different sizes using object recognition combined with the scale contained in the images. The main approach was to treat the coins independently of the underlying classification and analyse how an unsupervised method could group them. We later evaluated our results against the table provided and produced by the expert team. In addition, we have reviewed these expert classifications to improve them and provide a quality check, but also to get a better understanding of how the experts classify the coins, especially for those in poor condition. Additionally, we took a closer look at a single class and tried to identify coins fstruck with the different dies used. The phases of our work have been presented at CAA 2022 in Oxford and at CAA 2023 in Amsterdam.

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

Introduction

In our Classifications and Representations for Networks (ClaReNet) project, we are exploring and evaluating computer-based methods applied to three different Celtic coin series. One of them is the staters attributed to the Coriosolitae 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 (around 60,000 per coin face). Before the work began, Jersey Heritage invested a huge amount of labour and time (including 25 volunteers) to dismantle the hoard, take the photos and make a first identification of each coin. They also provided us with this data during the project. Our goal was to support the numismatic process, but also to show the potential of machine learning based methods. We aimed to evaluate different issues, from pre-sorting and classifying to the recognition of different dies. The main focus was on the use of unsupervised methods to also present how to approach an as yet unknown dataset, i.e. without any information other than the images themselves. Finally, we compared the results with the classification created by the Jersey team and also involved the expert into our process. This paper also presents tools, visualisations and extensions that have proven useful for the task, in communicating with the numismatists and integrating their opinion.

Data

The data set made available to us includes about 60,000 photos per coin face and contains various coins including staters, quarter staters and petit billions. In addition to the coins, the photos mostly include a scale and further information such as the assigned identification number (ID) or occasionally the assigned class. Both the coins and the photos are of mixed quality: there are broken, worn and corroded coins, as well as blurred, overexposed and underexposed photos (fig. 1). We focused on the staters, as they represent the largest part of the hoard (about 50,000) and their research is also more advanced. The staters are generally divided into six classes (I to VI, Appendix 1), which were originally formed on the basis of the obverse, or 'heads' face(Colbert de Beaulieu 1957), therefore we concentrated our work on the obverse too. The information we had to evaluate the process is:

- Staters have an average diameter of about 22 mm, quarter staters and petit billions about 13 mm.
- During the project, we received a table from the expert with information about the kind of coin (stater, quarter stater, …) and the assigned class.
- For one class of the staters (VI ~ 1300 images) there is an unpublished die study.

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

Overview of the pipeline

With the intention of extracting and analysing the staters, we followed a divide and conquer methodology. It aims at sequentially dividing the dataset into smaller batches in order to analyse each one more efficiently. Our pipeline thus includes the following steps:

1. Object Detection - For the rest of the process, the focus should be on the coin image, therefore the images need to be cropped. At the same time the size can be calculated by detecting of the scale, which allows a first sorting.

2. Unsupervised learning - In order to use only the images as input, methods were used that do not require any further domain knowledge. This step of the pipeline was repeated to extract groups of high similarity and with the goal of identifying groups similar to the expert's classification, while removing corroded and worn coins to eliminate the bias they can cause.

From here, the pipeline splits into two paths.

3a. Supervised learning - The result of step two was checked against the classification of the expert team and a classification model was trained to reassign the coins from the other batches. The domain expert was also involved in the process.

3b. In addition to sorting by class, methods for recognising and sorting by different matrices were tested. In this case, only the images of one class were used.

Object Detection

As already mentioned, the quality of the photos varies, but the format of the photos was consistent. For most of the photos four areas could be defined, although the last two may be missing: the assigned ID, the coin, the scale and the class (rare). 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 and could be found via the ID (fig. 2).

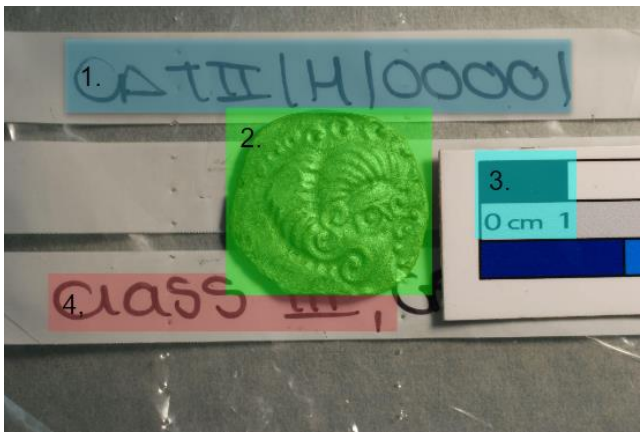

Figure 2 - The positions of the defined areas are consistent overall. (Photo: Jersey Heritage)

The implementation was done in a supervised process in which two classes, coin and scale, were defined, and a training and a test dataset were created. The position of the two objects on the images was consistent and the scale was also a less complex target due to its representation as a simple black box. This led us to select a relatively small data set in relation to the total number. 100 images were chosen as training data and 25 as test data; the images were chosen on a more random basis, but it was also important to cover observations such as small and large coins 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 gives a mean average precision of 95%. 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 manually re-measured 10 coins to get a feeling for the quality of the result, which turned out to be very accurate. Finally, we cropped the images and verified whether the coin is 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 is time-consuming and not essential for the further process. For the implementation, we used Tensorflow's Object Detection API[1] and its Model Zoo[2] in order to select a model architecture. We have not attempted to evaluate different architectures against each other in this paper. However, for our task we finally decided to use the CenterNet Hourglass104 512x512 by Duan et al. (2019) because it had a good balance between time and accuracy. In the Model Zoo, benchmarks are given that have been calculated on the COCO dataset[3], such as the mean average precision. These can be used as a guide when deciding on such an architecture.

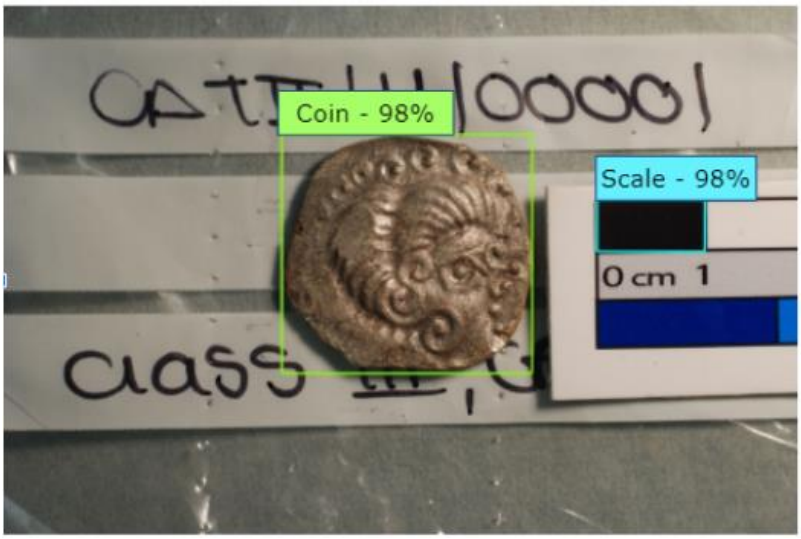

Figure 3 - Optimal Prediction of the model. Calculated values: height: 2.321cm, width: 2.194cm. (Photo: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

Once the model had been trained and tested, it was applied to the entire dataset. In the same prediction step, each image was cropped and the size was calculated. An optimal prediction is shown in Figure 3. The size calculation focuses on sorting the data set, not on an exact calculation of a minimum and maximum diameter. Therefore, we decided to take only two measurements: the width and height of the detected coin bounding box, dividing these two values by the width of the detected bounding box of the scale, which gives the value in centimetres. The cases where the size differed greatly from the others, and was therefore to be classified as an outlier, were considered separately. Figure 4 shows such a case. There are several reasons that could lead to an incorrect calculation. It can be seen that darker areas are classified as scale, which is not surprising when the target is a black box. In figure 4 there is also an area that achieved the same percentage as the scale itself, which can cause selection problems during the calculation. To counteract such cases, these and other suitable images (e.g. images with shadows) have been annotated, thus broadening the training base.

[1] https://github.com/tensorflow/models/tree/master/research/object_detection
[2] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md
[3] https://cocodataset.org/#home

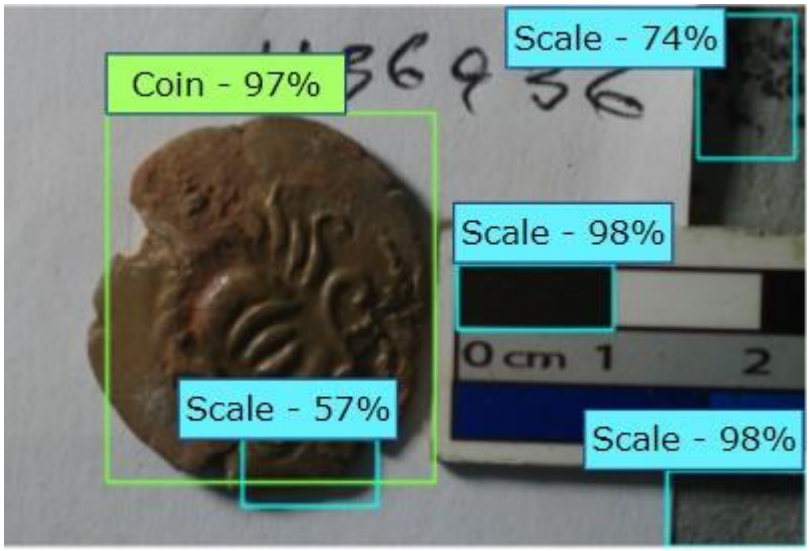

Figure 4 - Shadowy areas can lead to a wrong prediction by the model, resulting in an incorrect size calculation. (Photo: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The calculation was made for each photo, i.e. for both sides of the coin, resulting in four measurements for each coin. Comparing these also provides confirmation, as cases where the values are very different were examined more closely. The calculated values in centimetres can be viewed in a scatter plot in Figure 4. 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 red and large coins (probably staters) are marked 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 have chosen this as the separation. 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?'.

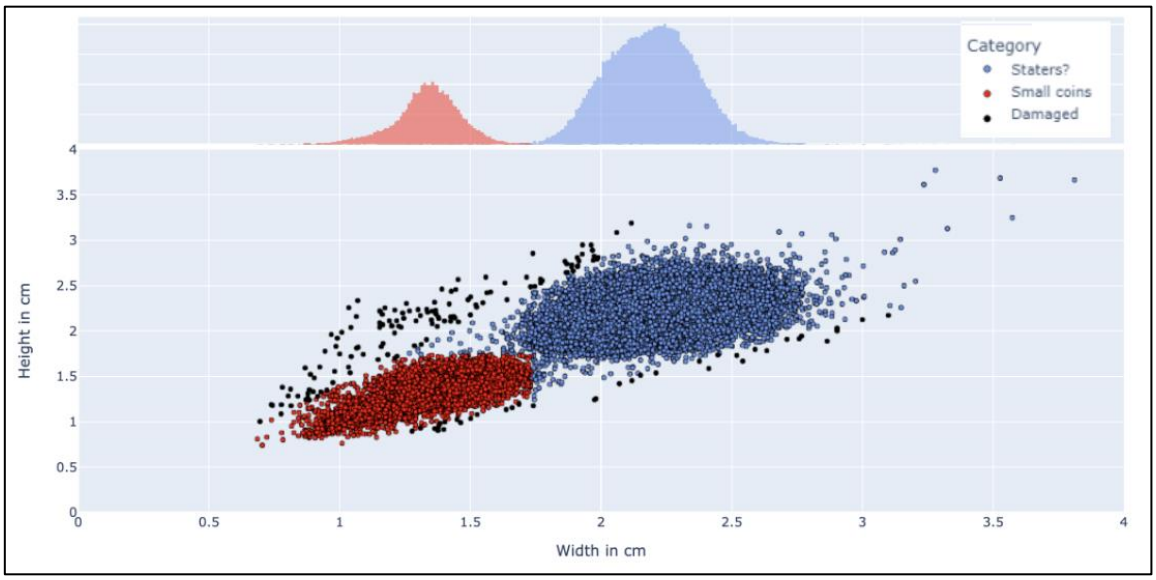

Figure 5 - Scatter plot of the approximated diameter. (Graphic: C. Deligio, Big Data Lab)

If we look at the peaks of the two point clouds, most of the elements are at around 1.3cm and 2.2cm. These two values correspond to the information provided by the expert and thus provide the first confirmation 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%). The latter contains photos where no scale was available or it was not detected, and therefore the calculation could not take place.

Unsupervised Learning

Taking our divide and conquer approach one step further, we selected the "Stater" group as the dataset for unsupervised learning. The main aim was to see if we can achieve a pre-sorting and to see how far it complies to the expert classification. Deep Learning and convolutional neural networks (CNNs) were the first choice for image classification, and a promising approach we found was the DeepCluster (Figure 6) method developed by Caron et al. (2018). This combines a convolutional neural network, which was what we wanted to use, with a clustering algorithm to train a CNN in an unsupervised manner. The idea in this approach is to use the generated clusters 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.

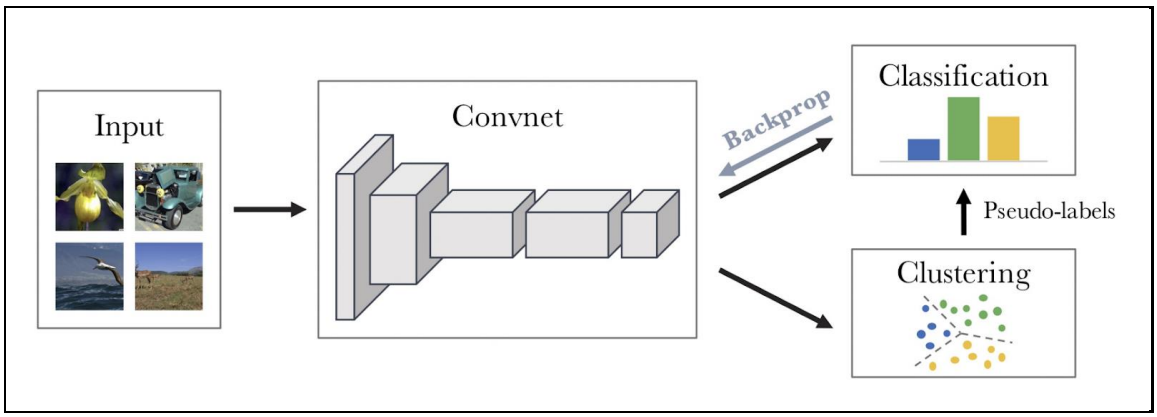

Figure 6 - DeepCluster implements a method of training a CNN in an unsupervised way. (Caron et al. 2018)

In the Paper by Caron et al. 2018, the method was used with the two CNN architectures alexnet and VGG16. We used the architecture with VGG16, and as the clustering algorithm k-Means. Thus, the required inputs were the images and the number of desired clusters (k). The choice of k was initially a challenge, for the paper recommends 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 is known (6), but since we wanted to analyse the dataset without using any 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 to see if they would be separated. For the evaluation of the resulting clusters, we avoided the use of additional information and performed it manually. Based on this manual evaluation, the following observations were made (fig. 7):

- Obverse and reverse were generally not mixed within the clusters → showing that the method is already working 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 and have been described as black boxes, so sometimes it is not clear how the results were obtained.

Figure 7 - Example of clusters generated. (Photos: Jersey Heritage)

Many clusters emerged that showed a strong similarity (based on our manual evaluation). Clusters with corroded and poorly preserved coins which had been identified 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 - or also clusters with a significant mix. In order to provide another degree of certainty, we focused on the coins at the centre of the size point cloud, i.e. 2.2cm +- 0.2cm (fig. 5). In this way, we wanted to ensure that only staters were present in the dataset (fig. 8). Also, we only used the obverse images.

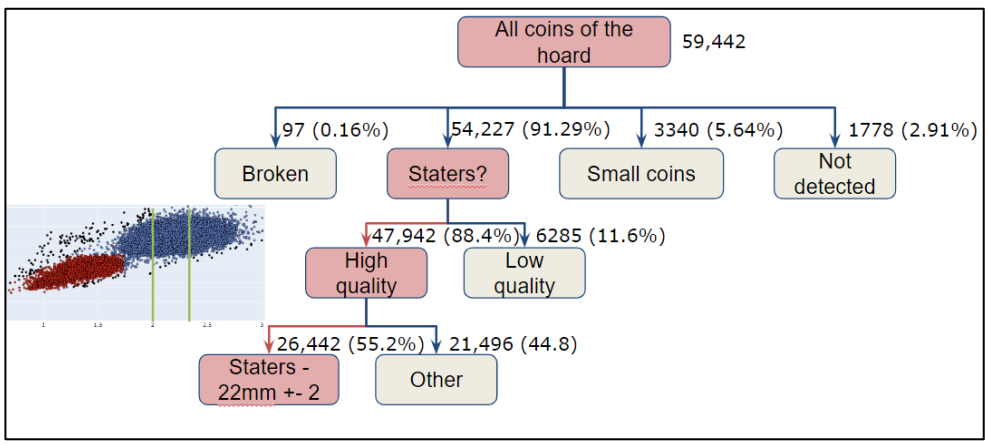

Figure 8 - Using the divide and conquer methodology, the data set could be divided step by step into more easily analysable parts. (Graphic: C. Deligio, Big Data Lab)

This new, reduced dataset (ca. 26,000 images) was now to be divided into clusters (25) once again, and when checking the results we evaluated them against the spreadsheet of the expert team. The first evaluation was to determine whether our selected dataset really did contain only staters. This could be confirmed. By comparing our results with the experts' spreadsheet, we could also calculate exactly which classes each cluster was composed of: the result was that 18 out of 25 clusters contained at least 70% of elements of only one single assigned class (15,063 images). Within this threshold, only 8% do not correspond to an assigned class (1208 images). The evaluation of the cluster is presented in Appendix 2. One drawback was that we did not manage to find any clusters of class VI coins which, with about 1300 occurrences, is by far the smallest class in the dataset. The images of class VI coins were mostly mixed into

clusters of class V or sometimes IV, which is probably a result of the similarity of the classes. In total, 13,855 coins (23% of the total data set) were confirmed by the comparison with the spreadsheet This data set, verified by two systems, by deep learning and the team's classification, now formed the basis for the supervised approach and verification of the coins previously excluded.

From unsupervised to supervised

The reduced and validated dataset was 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. As we did not want to lose class VI, we added the coins validated by the expert. As the unbalance was high, we rebalanced the dataset by downsampling to the smallest class. The goal was to increase the training material step by step. As a choice of CNN architecture, we started with VGG16. With the supervised model, we intended to evaluate and revise the other batches in our tree (fig.8). At the same time, we involved the domain expert, because the predictions had two outcomes: if the prediction matches the assigned one, it is basically confirmed. However, cases where the prediction differs are not automatically classified as false, but instead saved separately, to be reviewed. Confirmed coins were added to the training set so that the model constantly receives more material (fig. 9). This means we improved the model and checked the data quality at the same time.

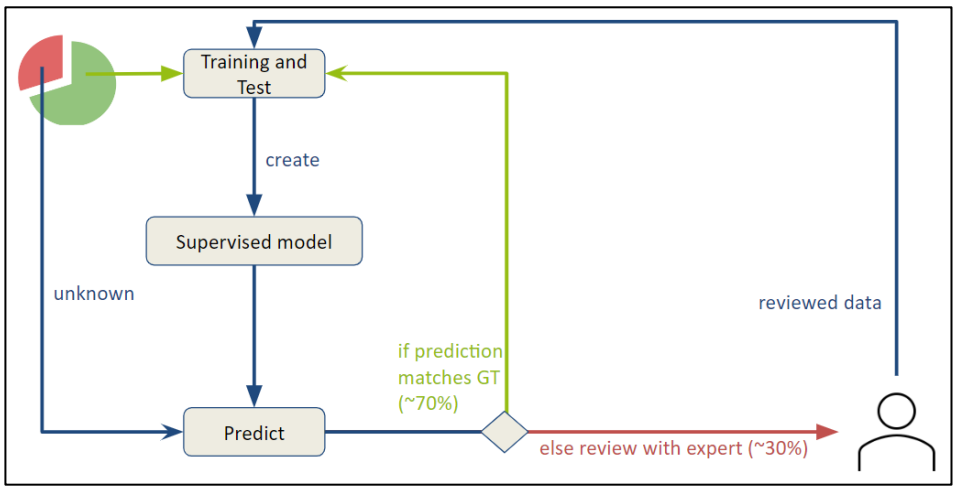

Figure 9 - The 13,855 images from the previous step were selected as the first training base (green). The 1208 wrongly selected ones as the first test set (red). The percentages are the results of the predictions on the test set. (Graphic: C. Deligio, Big Data Lab)

Whether the prediction or the given class was correct had to be decided by someone with domain knowledge, and therefore we involved the domain expert. We also used this step to design a kind of experiment. We created a list of coin images with two options, prediction and original assignment, but which were masked so that the expert did not know which was which, along with the ID. The domain expert could then choose one of the options or specify a new one. Comments were also welcome as especially in the case of difficult decisions, they helped us to understand them as non-experts. In our first test set (explained in Figure 9), the model was off by 30% (328 out of 1208 images). The review by the expert for these cases could be divided into four cases:

1. The assigned class was actually wrong and was improved by the model --> data quality improvement (115 cases - ~35%).
2. The determination 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 - ~4%)

The model mainly had problems distinguishing between classes IV and V, but this phenomenon also occurs with humans. We also distributed the list to our team (i.e. to non-experts) and asked them to fill it out. With the result that exactly these two classes also led to problems. In figure 10 we can see the evaluation of the 1208 images with the old classification (the one originally supplied) and with the expert's revised classification for these cases. The F1 metric on the revised classification increased by 10%. This shows above all that the performance of such a model cannot be calculated exclusively on the basis of metrics if the underlying data quality is not given.

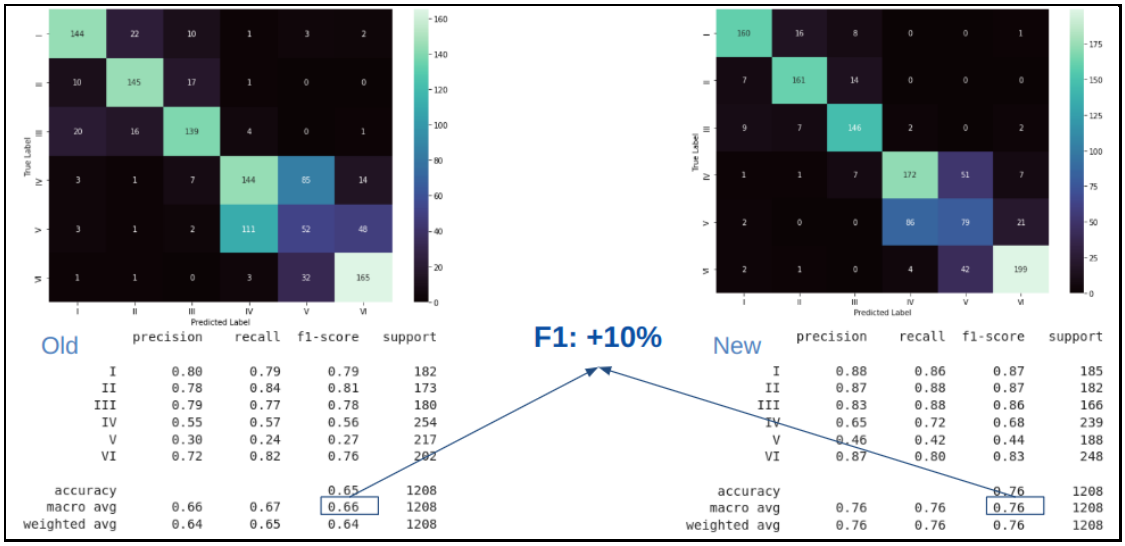

Figure 10 - Same predictions, different results. Comparison between two classifications (old vs revised). (Graphic: C. Deligio, Big Data Lab)

This process could now be repeated step by step for the remaining batches that had previously been sorted out. Ultimately, this could be used to improve the model and provide higher data quality.

An important question we also asked ourselves was how to deal with the coins defined by us as 'low quality', or to what extent such a model can help us. Using a random sample of 20 images per class from this set, we evaluated a small case study. It was of particular interest to see if the model that had been trained on very good images can be applied here. Our case study achieves an accuracy of 47%. Figure 11 shows the confusion matrix. The figure also visualises three examples with GradCam. It can be seen that correct regions (such as the hair or eye) are detected, but that there is also a bias due to the condition. The two images on the left are correctly classified by the model. Comparing the right-hand image with the images 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 showed the importance of the ground truth, and that it 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.

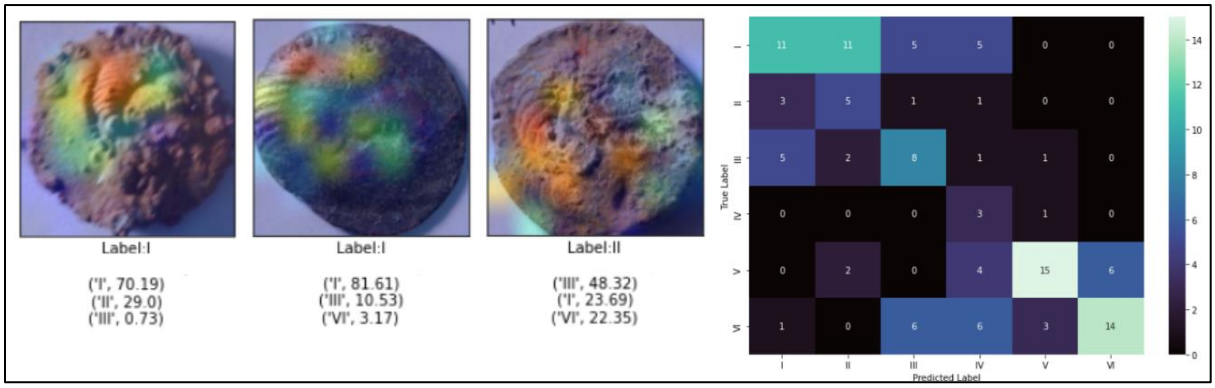

Figure 11 - On the left is the visualisation of the prediction with the top 3, on the right is the matrix of the 120 predictions. An F1 value of 44% and an accuracy of 47% were achieved. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

Visualisations and Augmentations

Figure 12 – Cutout random parts of the image. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The involvement of the expert also led to exciting insights, e.g. that the nose area plays a central role for him when assigning the class, but the visualisations of the predictions by GradCam, for example, did not reflect this focus. In the example of the coin in figure 13 (left), the right class is indeed in the top 3, but with only a 12.8% certainty. The important features, not always receive as much weight as desired. 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 helpful for our case. The cutout method involves hiding parts of the image, thus leading it to pay attention to several other areas (fig. 12). 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. 13b), we can see that certain areas have a stronger weighting, especially the eye region.

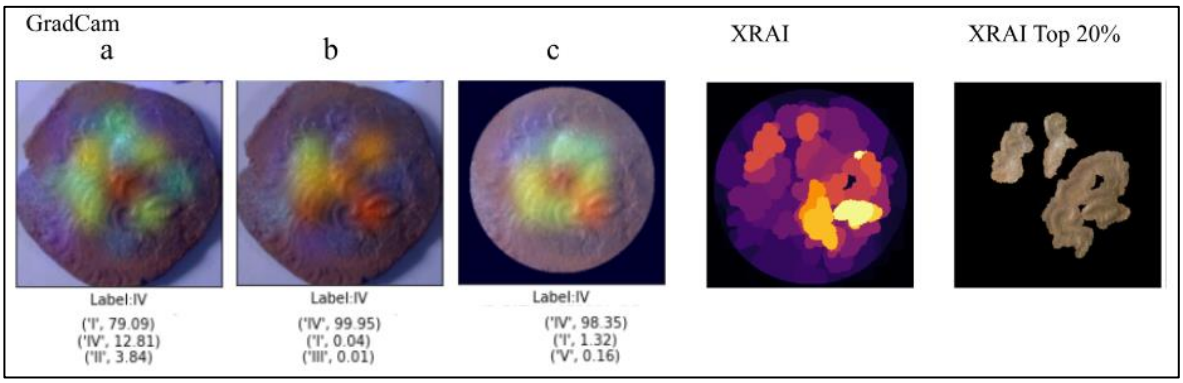

Figure 13 - GradCam visualisation and top 3 prediction of a model trained based on full coin (a), cutout (b), circle crop (c) images. Colour scale: blue (weak) - red (strong). On the right a visualisation with the use of XRAI (circle crop images) and cropping the top 20% area. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

Another augmentation, we simply call it the circle crop, was chosen because sometimes the edge of the coin, which can be very irregular, is focussed on and so can also cause noise. To counteract this, we

applied a simple circle crop oriented on the centre of the image to remove the edges. Figure 13 (c) shows that the focus of GradCam was very much in the centre, but less weighted compared to the cutout augmentation. In both cases you can also see that the class has been correctly identified.

As one can see, it is possible to direct the focus a little, and this offers the possibility of incorporating domain knowledge and to some extent also preferences. When it comes to explaining how a CNN works to a numismatic team, things can quickly get complicated. CNNs are difficult to understand due to their complexity and large number of parameters. To overcome this, there are various methods (SHAP, Lundberg and Lee 2017; LIME, Riberio et al. 2016; XRAI, Kapishnikov et al. 2019; and many more) of explanation, including visual ones, such as the GradCam method shown here. We recommend trying different methods and to communicate with the team to find a suitable one. In addition to GradCam, we also decided to use XRAI. 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[4] used also offered the option to extract the most important features instead of displaying a heat map, which was well received by the team and could be interpreted quickly

Implementing a die study

For a die study we dug further into a single class in order to try to distinguish individual coin dies within it. 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 applied methods. The recognition of dies brings another level of challenge for, although we are dealing with less data, the coins are very similar (same class) and there are many dies. For example, previously we had about 60,000 images with the goal of distinguishing six classes, but now we had only 1300 images with over 30 defined dies (based on the unpublished die study). For the implementation, we tested three methods against each other:

1. Reapplying DeepCluster.
2. Using our trained supervised model to extract features and then cluster them.
3. Algorithms comparing the key points in the image, which has also been successfully used in other publications.

The first two methods are similar in principle but differ in the trained CNN, in (1) it is trained from scratch using the DeepCluster algorithm, in (2) we used our trained model on the six classes and used it to extract features. (3) is a very different method to the ones used so far and will be discussed briefly here. The algorithms used are from the field of image matching. The best known are probably SIFT (Scale-invariant feature transform by Lowe, 2004), and SURF (Speeded Up Robust Features by Bay et al., 2006). As the first two are patented algorithms, a popular open source alternative is ORB (Oriented FAST and rotated BRIEF by Rublee et al., 2011). Such algorithms have also been successfully applied to ancient coins in various publications (Kampel and Zaharieva 2008; Taylor 2020; Heinecke et al. 2021). For our procedure we used ORB. There are also some pre-processing steps that had a positive effect on the results in terms of reducing bias (like scratches from the usage of the coin). The images were converted to greyscale in order to avoid a colour bias. The images were also 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 (fig. 14 shows the output).

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

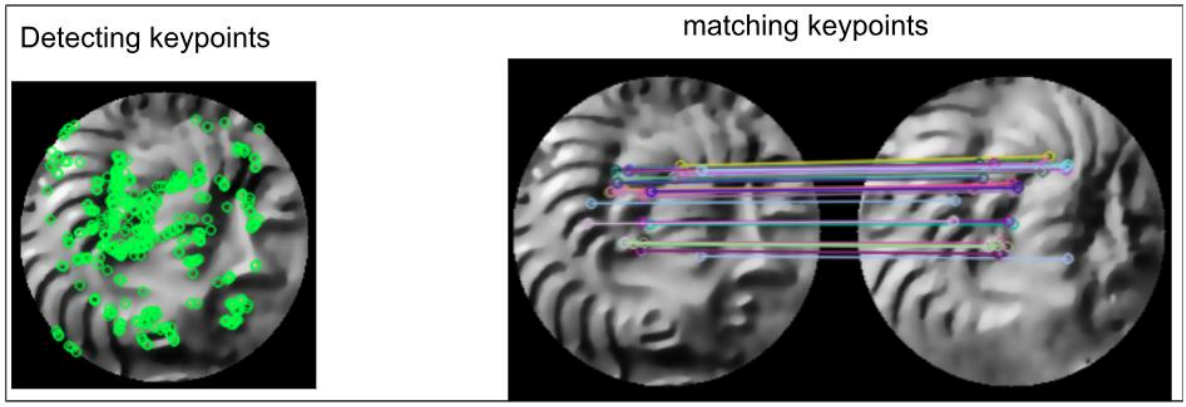

Figure 14 - On the left, an image with detected keypoints. On the right, an example of two supposedly identical die pieces (according to the GT) and their matches. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

The process is then as follows, the first step consists of the key point detection algorithm and matching the key points between two images. This comparison is carried out in pairs between all images and the matches found are captured as a vector for each image (resulting in an n x n matrix). For the exact method of calculating the key points, we recommend the ORB publication (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 all used as input to a clustering algorithm, in this case we used hierarchical clustering. We used the Orange Data Mining tool for calculating distances (based on Spearman distance metric), clustering and visualisation. Figure 15 shows the visualisation of the clustering as a dendrogram, in which we compare the result of the clustering with the existing die study of the expert. In order to evaluate the methods equally, there are now various possibilities; we decided to evaluate them all with the same distance value. We started with the image matching method (3) and the value 0.3 proved to be optimal and 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 identified, the mean value of all clusters and the total number of clusters formed.

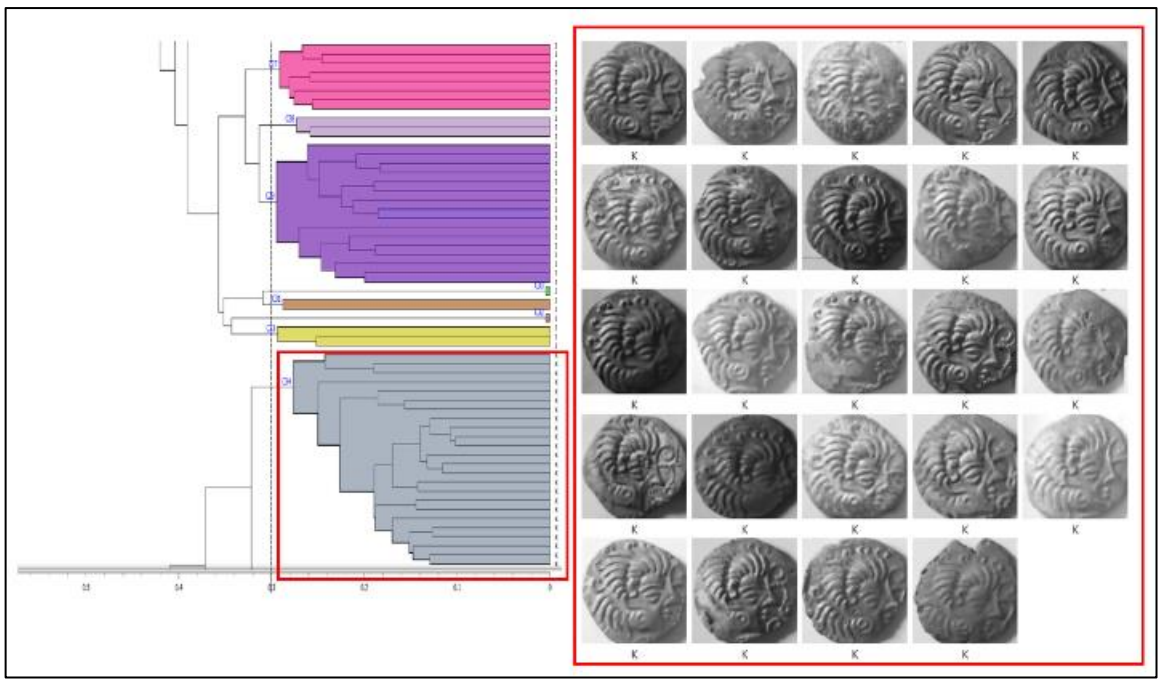

Figure 15 - On the left, part of the dendrogram. On the right, an overview of a cluster successfully containing the coins of one die. (Photos: Jersey Heritage. Graphic: C. Deligio, Big Data Lab)

It is noticeable that the DeepCluster method performs better with a larger amount of data and in this case struggled with the relatively high number of classes on this small amount of images. The second method, our supervised trained model, performed slightly better. The advantage was 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 clusters had at least 70% coins from the same die. Furthermore, 194 of them had only coins of exactly one die. The 194 cluster cover 489 coins and that was 40% of the dataset considered here.

Tabel 1 - To compare the methods we used the same threshold (0.3), we calculated two values to get an impression of the performance: Highest matching cluster with gt and the mean over 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 a case study of a new find, with no information available at the outset. With the first step of object detection, it was possible to automatically crop the images and, with the scale, calculate the size of the coins and perform some sorting, which helped to identify the staters. The next step of unsupervised learning was to see how far the dataset could be sorted or grouped, and whether it could even match the classification of the expert. By repeatedly applying the method with manual evaluation and then combining it with the size calculation, it was possible to extract a dataset containing only staters and to extract about 14,000 coins, which were correctly grouped according to the provided spreadsheet.

Re-evaluating the data with a supervised method leads to an improvement in data quality and in critical cases the AI's and the expert's decision can be compared . It is necessary to involve the domain expert in the process.

Specific additions can influence the choice of features and bring a model closer to the expert's expectations.

Sometimes it is also useful to use have a look on other algorithms (in terms of complexity and computing power required), in our case image matching (ORB), especially with small datasets (many classes, high similarity of data), as is shown by the positive results of our die study and of tests presented in other publications. Selecting the right approach and algorithm is still a challenge, also for IT experts. This is partly due to the fact that each approach is also accompanied by various subtasks (e.g. different ways for

401 preprocessing or augmentation) and possibilities for fine tuning (e.g. the number of clusters,
402 hyperparameter settings, choosing the best loss function).
403
404 The results show that semi-automatic support can be helpful in conducting sorting, classification, or even
405 a die study. A tandem approach where domain and IT experts work closely together will probably have the
406 best success rate.

Data, scripts, code, and supplementary information availability

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

Dataset used in this paper: https://doi.org/10.34780/kzw0-r608

Official website of the dataset: https://catalogue.jerseyheritage.org/categories/celtic-coin-hoard/

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 that they comply with the PCI rule of having no financial conflicts of interest in relation to the content of the article.

Funding

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

References

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.2112.0290

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.

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>

478

Appendix

479

480 **Appendix 1** - The six classes of staters as defined by numismatists. (Photos: Jersey Heritage)
481

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 frontal images with k=25. Values above the threshold of 0.7 are shown in green.