2014 22nd International Conference on Pattern Recognition

# Ancient Coin Recognition Based on Spatial Coding

Jongpil Kim and Vladimir Pavlovic
Rutgers, The State University of New Jersey
110 Frelinghuysen Road
Piscataway, NJ 08854, USA
Email: {jpkim, vladimir}@cs.rutgers.edu

***Abstract*—Roman coins play an important role to understand the Roman empire because they convey rich information about key historical events of the time. Moreover, as large amounts of coins are daily traded over the Internet, it becomes necessary to develop automatic coin recognition systems to prevent illegal trades. In this paper, we propose an automatic recognition method for ancient Roman coins. The proposed method exploits the structure of the coin by using a spatially local coding method. Results show that the proposed method outperforms traditional rigid spatial structure models such as the spatial pyramid.**

## I. Introduction

A coin is usually a flat piece of metal issued by governmental authority as a medium of exchange. It has been produced in large quantities to facilitate trade from the ancient history to the present. Along with the trading purpose, the Roman empire knew how to effectively use the coin as their political propaganda. The ancient Roman coins were widely used to convey the achievements of Roman emperors to the public. They were also served to spread messages of changing policies or merits through the empire. By engraving portraits on the coins, the Roman emperors also could show themselves to the entire empire. In short, the coins were the newspaper of the Roman empire. In this way, the Roman coins are always connected to historical events and Roman imperial propaganda. Therefore, understanding the ancient Roman coins could serve as references to understand the Roman empire.

The Roman coins have been collected by people as a hobby because of not only their bullion value but also their artistic value. Moreover, as the coins were massively produced, there are huge numbers of the Roman coins so that they are not rare like other antiquities. In addition, new Roman coins are daily excavated, make themselves affordable to collect and become popular for non-academic enthusiasts. Most of the coins today, in fact, reside in private collections [1].

Because the coin market is very active, a lot of coins are traded every day, mostly over the Internet [1]. But ancient coins are also becoming subject to a very large illicit trade [2]. A traditional way to detect the illegal traffic of the ancient coins is to periodically and manually search catalogue, dealers or the Internet by authority forces. But those methods have limits to prevent the illegal trade because manual process is too slow to cover all the trades. Therefore, there is a need to develop a both reliable and automatic method to recognize the coins.

There are tens of thousands of typologies that could be used to classify Roman coins [1], [3]. Therefore those who do not have knowledge and experience cannot classify them without the help of experts or automatic classifiers. In this paper, we focus on the recognition of the Roman emperors on the Roman imperial coins. Specifically, for a given coin image, we proposed an automatic method to recognize who is on the coin.

(a) Vespasian

(b) Vitellius

Fig. 1: Inter-class similarity in the ancient Roman coins. Vespasian looks similar to Vitellius.

Inter-class similarity and intra-class similarity are two challenges to recognize the ancient Roman coins. For the inter-class similarity, different emperors share similar appearance as shown in Figure 1 and Figure 10. There are several reasons for the similar appearance: familiar relatedness, engraver's lack of knowledge for the emperor's image or abstraction, or using the same template for different emperors. Another aspect of the coin recognition challenge is the intra-class dis-similarity as shown in Figure 2. There may exist a large variation within the same class. On a very basic level, the direction of the emperor's face varies over the coins: some emperors look left and the others look right without any specific rule as shown in Figure 2.

Several works have proposed to recognize the coins [4], [5], [1], [6], [2], [7] in the computer vision community. They represent the coin image as low level visual features, ignoring the structure of the coin [5], [4]. Arandjelović [1] introduces a directional kernel which indirectly uses the structure information but did not explicitly facilitate the use of the spatial structure of the coin.

In this paper, we address a problem of automatically recognizing ancient Roman coins, while leveraging their spatial structure and without focusing on the textual transcripts. The ancient Roman coins have regular structure: the coin is round, the location of the emperor is at the center of the coin, the emperors share common aspects all over the coins, and so on. We propose a framework to leverage the coin structure to improve the recognition accuracy. To the best of our knowledge, this is the first paper which uses the structure of the

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.64

Fig. 2: The emperors on the five coins are the same, Nero. But there is variations on the shapes. In particular, one face looks left while the others look right.

Roman coin for the recognition. Spatial pyramid models [8] are usually used to encode spatial location information by defining artificial boundaries (*e.g.*, rectangular grid). On the other hand, the proposed model directly handles spatial locations without the artificial boundaries. The experimental results show that the proposed coding method performs better than the spatial pyramid models.

For this research we have collected a new ancient Roman Imperial coin dataset, as a part of our contribution to the computer vision community. All the coins are annotated and consist of high-quality images.

This paper is organized as follows: Related work is summarized in Section II. In Section III, we describe a new coin dataset and outline basic coin preprocessing steps. Then, we explain our proposed method in Section IV. Lastly, we show experimental results in Section V and make conclusion in Section VI.

## II. Related Work

Several methods for recognition and analysis of modern coins rely on gradient information based approaches [4], [5] and eigenspace decompositions [6]. But none of them are adequate for the ancient coin classification because the ancient coins are too often in very poor conditions, common recognition algorithms can easily fail [2]. In [9], SIFT descriptors [10] are used to obtain 90% classification accuracy for 390 coin images where there are only 3 classes. A directional histogram to consider orientations of pixels was proposed in [1] for the ancient Roman coin classification. The method proposed in [7] is based on a dense correspondence search between coin images. Unlike the previous works, the proposed method explicitly considers the spatial structure of the coin.

The coin recognition problem can be considered as the face recognition in terms of recognizing an Emperor's face on the coin. Many methods have been developed for recognition of real face images. However, the use of such methods faces significant challenges when applied to ancient coins. In terms of its gradient/edge content, a typical critical feature used in face recognition, most ancient coins display vastly different statistics from photographed faces. This aspect is further aggregated by the fact that many coins are old, worn-out and damaged. For example, Figure 3 shows different HoG [11] distributions between the coins and the real faces. Tzimiropoulos *et al.* [12] proposed a method to learn subspace from image gradient orientations (IGO subspace learning) for appearance-based face recognition that was shown to be very robust to different types of image noise. The advantage of using the IGO subspace learning algorithm is that the cosine distance measure of the algorithm can cancel out outliers or noise caused by occlusion or illumination changes. However, the IGO-algorithm is very

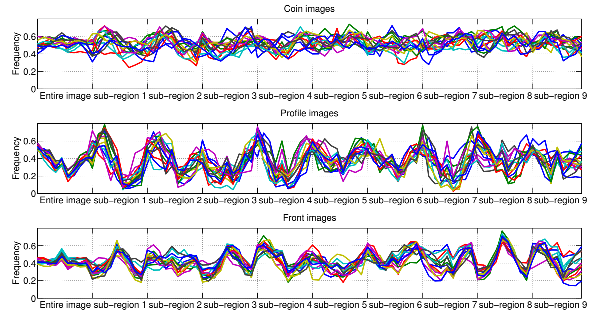



Fig. 3: HoG Distributions for coin and real face images. We extract the HoG descriptors from the entire region and $3 \times 3$ sub-regions. Each line represents the HoG descriptor distribution of one sample. Top: coin images, Middle: profile images, Bottom: frontal face images. The real faces show regular patterns while the coins show different distributions.

sensitive to the alignment, requiring exactly aligned images which require an additional huge amount of human effort for the coin images. On the other hand, the proposed method is a fully automatic one not requiring any human intervention.

## III. Coin Dataset Generation

In this section, we describe how to collect coin images, perform background removal and correct face direction.

### A. Coin Data Collection

We collect ancient Roman coin images from a numismatic web site. Each coin has a high resolution image (approximately $350 \times 350$ pixels jpeg image). Among the collected coin images, we found that some of the coins are hard to recognize because they are rusty and severely damaged. As we are dealing with the problem of recognizing the Roman emperors, a coin that is severely damaged or hard to recognize who is on the coin is discarded. After removing such worn-out coins, we select emperors who appeared more than 10 times in the dataset. Finally, we arrive at 2815 coins with 15 emperors. The sample images and the frequencies are depicted in Figure 10 and Figure 4, respectively. All images in the dataset are ancient Roman Imperial coins dated from 27 BC to 355 AD. In this paper, we consider only the observe(front) of the Roman coin because the emperor is engraved in the observe and the reverse usually shows various non-face symbols.

### B. Background Removal

There are many well-known methods such as GrabCut [13] to separate foreground from background. In this paper, we observe that the Canny edge detector [14] can find the area of the coin in the image. Figure 5 depicts the foreground separation process. First, the Canny edge detector finds edges for a given coin image. Figure 5b shows the edge detection results. Then we fill out the inside of the outline, having the mask for the coin area as shown in Figure 5c.

After finding the mask of the foreground, we add 5% of the estimated diameter of the mask as a padding. The diameter is estimated by averaging the largest horizontal and vertical

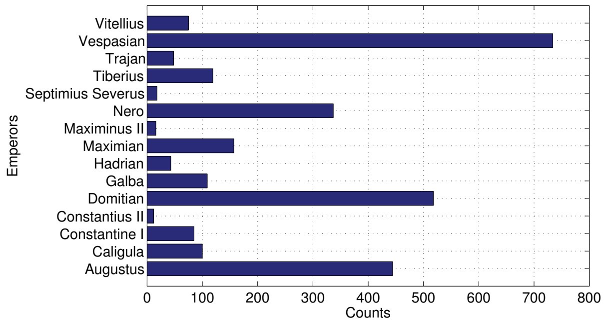



Fig. 4: Histogram of the number of coins in each of the 15 Roman emperors in our dataset.

(a) Original coin image (b) Canny edge detection (c) Detected coin area

Fig. 5: Foreground detection in the coin image.

length of the mask. We then resize the coin image to a fixed $350 \times 350$ pixels.

### *C. Face Direction*

As we can see in Figure 1 and Figure 2, different emperors look at different directions without any specific rule. Therefore, it becomes necessary to *automatically* make all the emperor look at the same direction otherwise we have to consider two different coin structures. Figure 6 depicts a fast and effective method to determine the direction of the face on the coin. For each coin image, we divide it into left and right areas with an overlapping part (red for left, blue for right and magenta for overlap). We put the overlap for the case when a face area on the coin is severely biased to the left or to the right. We extract visual features, constructing histograms $h_{src}^{left}$ and $h_{src}^{right}$ (resp. $h_{trg}^{left}$ and $h_{trg}^{right}$) from the left and right regions of the source image (resp. the left and right regions of the target image). Then, we measure two distances between the source and the target images and between the source and the flipped target images using the histograms. If the former is larger than the latter, we simply flip the target image otherwise do nothing. Based on this method, we determine the face direction of all the coin images in the dataset using the algorithm depicted in Figure 7. The algorithm assumes that we are given a single image as which we make all the other coin images look at the same direction. From a set the single image, we increase the size of the set by iteratively adding images. The algorithm performs the majority vote to determine the direction which prevents the algorithm from making a wrong decision for a biased coin. Using this method, we can determine the direction of the face with 100% accuracy.

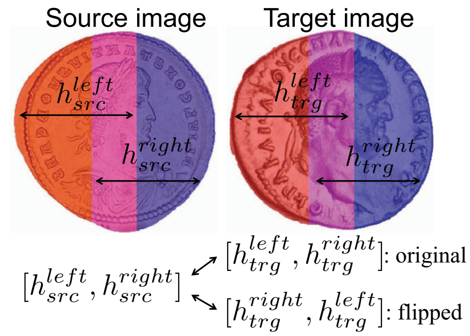



Fig. 6: A method to determine whether two emperors look at the same direction or not.

Input: $\{I_1, \ldots, I_N\}$, Output: $S$ (face directions are corrected)
$S = \{I_1\}$
**foreach** $i \in \{2, 3, \ldots, N\}$
  **foreach** $j \in \{1, \ldots, i-1\}$
    $d(j) = D([h_j^{left}, h_j^{right}], [h_i^{left}, h_i^{right}]) - D([h_j^{left}, h_j^{right}], [h_i^{right}, h_i^{left}]) < 0$
  **end**
  **if** majority_vote($\{d_1, \ldots, d_{i-1}\}$ == 1
    Do not flip $I_i$
  **else**
    Flip $I_i$
  **end**
  $S = S \cup \{I_i\}$
**end**

Fig. 7: Algorithm to make all the emperors look at the same direction. $D(\cdot)$ is a measure to calculate a distance between two histograms such as the $\chi^2$ distance.

## IV. Proposed Method

In this section, we explain a baseline approach. Then we propose a visual-spatial feature to recognize the ancient Roman coins.

### *A. Baseline Approach*

We use the bag-of-words model for constructing the visual histograms. The model 1) selects a set of key-points, 2) extracts visual descriptors on the set of key-points, 3) the descriptors are quantized into a visual codebook and 4) an input image is represented as a histogram of the codewords in the codebook.

There are a number of ways to find interest points including dense regular grid [15] and difference-of-Gaussian (DoG) peaks [10]. We choose the dense grid sampling because it shows better performance than the DoG sampling method. The SIFT descriptor [10] is used to extract the visual descriptors on the grid points. Then we use $k$-means clustering to build the visual codebook, generating the histograms for the coin images using the visual codebook.

The dense sampling gives the same weight to all the key-points, ignoring the spatial location in the image. The coin has the regular structure, but the dense sampling makes no use of this information. To overcome this limitation, we follow the spatial pyramid models widely used in computer vision

Fig. 8: Left: grid sub-regions. Right: the polar coordinate system. We build a histogram on each region and concatenate all the histograms to represent one image.

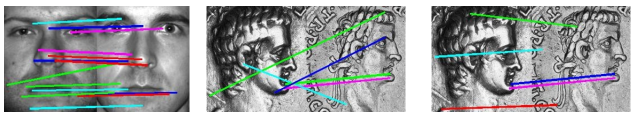


Fig. 9: SIFT matching results on the real faces and the coin images. Left: matching using SIFT descriptors on the real faces. Middle: matching using SIFT descriptors on the coin images. Right: matching using the proposed method.

area [8], [16]. Sampling on predefined sub-regions of an image (*e.g.*, $1 \times 1, 2 \times 2, 4 \times 4$) has been suggested to improve the performance. We also use the polar coordinate system to take an advantage of round shapes of the coin and the face of the emperor. Figure 8 depicts the examples of the grid sub-regions and the polar coordinate system on the coin image. In the polar coordinate system, we place the center of the system at the center of the coin and the radius of the inner circle is set to cover the face of the emperor. One can find that the inner circle separates the face from the legend of the coin.

### B. Spatial Coding for Coin Recognition

Edges and shapes on the coin images are weaker than those on the real face images as the coins are old and contaminated by damages, rust, worn-out. More importantly, the bas-relief ambiguity [17] also makes it difficult to recognize the shape of the coin. We illustrate the difficulty using the SIFT matching [10] in Figure 9. For a fair comparison, we crop the regions to contain only faces. The SIFT matching can find matched points for the real human faces as shown in the left panel of Figure 9. However, the matching method dose not find adequate points properly on the coin image as shown in the middle panel of Figure 9. The bas-relief ambiguity of the coin misleads the matching method. For example, a point on the chin is matched to a point on the eye (blue line).

To overcome the aforementioned problem, we propose a method to use the spatial structure of the coin. McCann and Lowe [18] proposed a spatially local coding method to handles the spatial locality. We adopt the local coding method to encode the coin image to use the spatial locality information without relying on the spatial pyramid models. We expect the spatial locality to reduce the ambiguity of the coin images.

Let assume that a pixel $p$ of an image is encoded as

$$\phi(p) = [v_1, v_2, \ldots, v_d]^\top ,$$

where $v_i$ is the $i$th visual descriptor (*e.g.*, SIFT) and then the pixel is represented as a $d$-dimensional vector. We augment the representation by adding the location of $p$ as

$$\phi^l(p) = [\phi(p), \lambda x, \lambda y] = [v_1, v_2, \ldots, v_d, \lambda x, \lambda y]^\top ,$$

where $x$ and $y$ are the two normalized coordinates of the pixel in the image and $\lambda \in \mathbb{R}$ is a parameter to control the importance of the location.

After extracting the augmented features from the images, we run the $k$-means clustering algorithm to construct the codebook by minimizing

$$\begin{aligned} L(D, \mathbf{a}_i) &= \sum_{i \in Tr} \left\| \phi^l(p_i) - D\mathbf{a}_i \right\|_2^2 \\ &= \sum_{i \in Tr} \left( \left\| \phi(p_i) - D^v \mathbf{a}_i \right\|_2^2 + \left\| \lambda [x_i, y_i]^\top - D^l \mathbf{a}_i \right\|_2^2 \right) \quad (1) \\ & \quad s.t.\ |\mathbf{a}_i|_1 = 1, |\mathbf{a}_i|_0 = 1 \text{ and } \mathbf{a}_i \succeq 0 \end{aligned}$$

where the superscripts $l$ and $v$ represent the location descriptor and the visual descriptor, respectively, $Tr$ is the set of indices for the training data, $D = [D^v; D^l], D \in \mathbb{R}^{(d+2) \times K}, D^v \in \mathbb{R}^{d \times K}, D^l \in \mathbb{R}^{2 \times K}$ is the codebook and $\mathbf{a}_i \in \mathbb{R}^{K \times 1}$ is the codeword for $\phi^l(p_i)$. $L_0$ and $L_1$ norms are used for the hard assignment, *i.e.*, only one element of $\mathbf{a}_i$ is 1 and all the others are 0. For a new feature vector, we assign it to the nearest codeword in the codebook. Equation 1 is reverted to the typical bag-of-feature representation by setting $\lambda = 0$.

Minimization of Equation 1 seeks to produce codes $\mathbf{a}_i$ such that not only visually similar descriptors but also spatially close descriptors belong to the same cluster. In this sense, it is close to Locality-constrained Linear Coding (LLC) [19]. But the proposed method considers the locality in the pixel space while LLC does in the descriptor space. By using the location information in the pixel space we can facilitate the use of the spatial structure of the coin. The right panel of Figure 9 shows the effectiveness of the augmented representation as the matching algorithm can find the adequate points on the coin images. To archieve good classification performance, similar descriptors should produce similar codes [19]. In the proposed method, we extend this idea into the pixel space so that the both visually and spatially similar parts in the coins produce similar codes. Experimental results show that the proposed method successfully contributes to the recognition performance.

## V. Experimental Settings and Results

### A. Experimental Settings

We examine the performance of the proposed method on the coin dataset which we constructed. The coin dataset consists of 2815 images with 15 Roman emperors. For the purpose of the evaluation, the coin dataset is randomly partitioned into 5 equal size subgroups. Each subgroup keeps the same emperor ratio as the total coin dataset. Then, we use 4 subgroups as training data and 1 subgroup as test data. Experiments are repeated 5 times so that each of 5 subgroups becomes the test data, and we report the average of the 5 outputs as a final output.

We extract the SIFT descriptors from the image as visual features in the dense manner. The location of the descriptor is normalized by the size of the image, ranging from 0 to 1. We obtain a codebook with $K$ words by $k$-means clustering on the extracted features and then the representation of the image becomes a $K$-dimensional vector. In this paper, we set $K$ to 2000. $K$ was chosen because it shows the best performance. In fact we will investigate how $K$ impacts prediction accuracy.

Fig. 11: Left: results of $k$-means clustering, where each green dot represents a location of each codeword. Right: red dots for locations of the most discriminative codewords and blue dots for locations of the least discriminative codewords

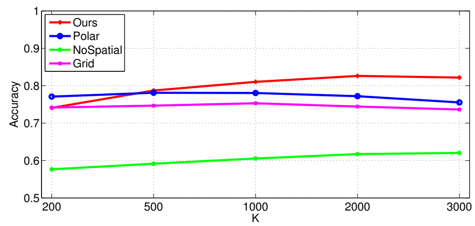



Fig. 12: Recognition accuracies for various $K$. The performance goes down when $K$ is larger than 2000.

We use the $\chi^2$ distance to calculate the distance between two images. The $\chi^2$ histogram distance is based on the $\chi^2$ test-statistic to test the fit between a distribution and observed frequencies and has shown successful performance in the object categories classifications [20], [21].

The multi-class SVM is used for training and prediction with the RBF-$\chi^2$ kernel. All parameters are determined by the 5-fold cross validation. For the evaluation, we calculate the average number of correctly classified test samples.

### B. Experimental Results

To set the baseline performance, we use three different methods: `NoSpatial` does not use the spatial pyramid model, `Grid` uses the grid sub-regions and `Polar` the polar coordinate system. We also implemented a directional kernel (`DK`) method proposed in [1].

Figure 11 depicts the results of the $k$-means clustering. In the left panel, each green dot represents the location of the codeword in the codebook (*i.e.*, $D^l$ in Equation 1). The codewords located at the face area and the legend area where there exists strong edges while smooth areas have smaller number of the codewords. As we use the legend which provides rich information about the coin, the proposed method implicitly uses contextual information. In this sense, the coin recognition is different from the face recognition where there is no contextual information.

To measure the importance of the codewords in terms of the location, we follow a method described in Section 3.4 in [22]: 1) we first select a set of the codewords that are spatially contained in a circle, 2) we zero features associated with the selected codewords, 3) we run the SVM trained as above with the newly created features, 4) we repeat this process. The prediction difference between the original features and the new features implies the contribution of the selected codewords: the larger the difference is, the more important the set of the codewords is. The red dots and the blue dots in the right panel of Figure 11 show two sets of most important and two sets of least important regions, respectively. We can find the the forehead area is the most distinctive because different emperors have different hair ornaments and hairstyles. On the other hand, the legend area is the least discriminative because gradient-based features (*e.g.*, SIFT) is insufficient to extract the contextual information.

TABLE I: Coin recognition accuracy (%)

| | `NoSpatial` | `Grid` | `Polar` | `DK` [1] | ours |
|---|---|---|---|---|---|
| ACC | 61.7 ($\pm$ 2.0) | 74.4 ($\pm$0.8) | 77.2 ($\pm$0.6) | 33.0 ($\pm$ 3.0) | **82.3** ($\pm$1.3) |

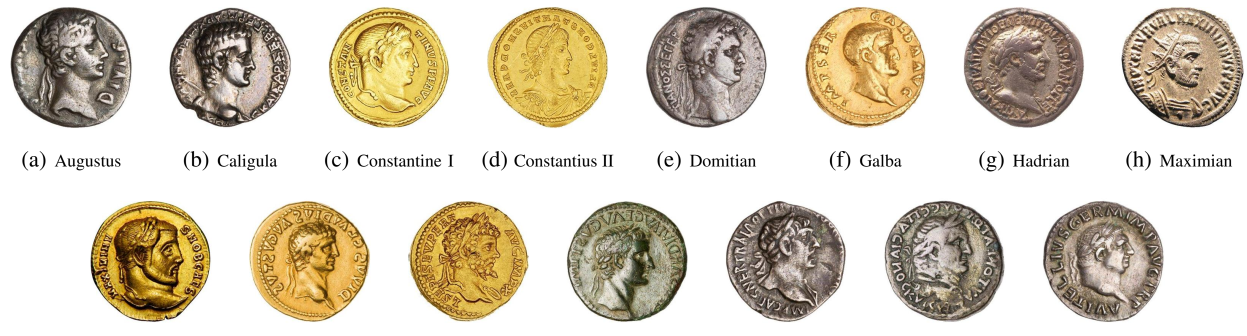
The recognition accuracies are summarized in Table I. The methods using the spatial information perform much better than `NoSpatial`. In particular, the proposed method is about 21% better than `NoSpatial`, implying that the spatial location plays an important role in the recognition. `Polar` performs better than `Grid` because the circular shape of the polar coordinate system conforms to the shape of the coin. The proposed method gains 5% improvement over `Polar` without pre-defined spatial pyramids such as the grid sub-regions and the polar coordinate system This result implies that by directly encoding the location, the spatial code used in the proposed method has more information about the structure of the coin than any other spatial pyramid models. `DK` shows the worst performance because it depends on the DoG sampling method which performs worse than the dense sampling in the collected coin dataset.

Figure 12 shows how the number of codewords in the dictionary impacts the recognition accuracy. If $K$ is less than 500, `Polar` performs better than the proposed method because 500 words is insufficient to encode both visual and spatial locality for the proposed method. However, as $K$ increases, the proposed method outperforms `Polar`. In fact, the proposed method produces more compact representation than `Polar` because `Polar` concatenates $K$-dimensional vectors from a set of regions (*e.g.* the total dimension becomes $K \times r$ where $r$ is the number of the regions).

Figure 13 depicts the confusion matrices of `NoSpatial`, `Polar` and the proposed method. The proposed method successfully improves the recognition accuracies all over the 15 classes. `Maximian` shows the best performance because he has a discriminative shape, spiky hair as shown in Figure 10h. `Constantinus II` and `Maximinus II` show the worst performance because the number of training images for them are relatively lower than the others and their appearances are similar to the others so they are easy to be confused.

## VI. Conclusion

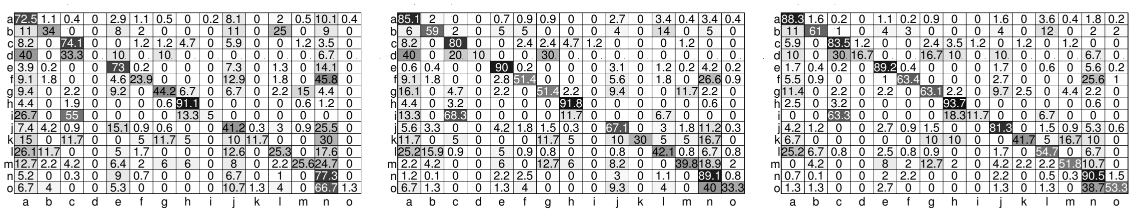
We proposed an automatic method to recognize the ancient Roman coins. The proposed method directly handles the spatial location of the visual feature to encode the image. Therefore, unlike the spatial pyramid models, our method does not need to find out the adequate structure of the pyramid. The experiments show that the propose method outperforms the other methods based on the spatial pyramid models.



(a) Augustus (b) Caligula (c) Constantine I (d) Constantius II (e) Domitian (f) Galba (g) Hadrian (h) Maximian

(i) Maximinus II (j) Nero (k) Severus (l) Tiberius (m) Trajan (n) Vespasian (o) Vitellius

Fig. 10: An example observe image of a coin for each of the 15 classes in the dataset.



(a) `NoSpatial` (b) `Polar` (c) Proposed method

Fig. 13: Confusion matrices for `NoSpatial`, `Polar` and the proposed method.

## References


[1] O. Arandjelović, "Automatic attribution of ancient roman imperial coins." in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[2] M. Zaharieva, M. Kampel, and S. Zambanini, "Image based recognition of ancient coins," in *Conference on Computer Analysis of Images and Patterns (CAIP)*, 2007.

[3] H. Cohen, *Description historique des monnaies frappées sous l'empire romain*, 2nd ed. Paris, France, 1880-1892, no. Vol I-VIII.

[4] L. J. P. van der Maaten and P. J. Boon, "Coin-o-matic: A fast system for reliable coin classification," in *MUSCLE CIS Coin Competition Workshop*, 2006.

[5] M. Nölle, H. Penz, M. Rubik, K. Mayer, I. Hollaender, and R. Granec, "Dagobert - a new coin recognition and sorting system," in *Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2003.

[6] R. Huber, H. Ramoser, K. Mayer, H. Penz, and M. Rubik, "Classification of coins using an eigenspace approach," *Pattern Recognition Letters (PATREC)*, vol. 26, pp. 61–75, 2005.

[7] S. Zambanini and M. Kampel, "Coarse-to-fine correspondence search for classifying ancient coins," in *Asian Conference on Computer Vision (ACCV)*, 2012.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.

[9] M. Kampel and M. Zaharieva, "Recognizing ancient coins based on local features," in *International Symposium on Advances in Visual Computing (ISVC)*, 2008.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, pp. 91–110, 2004.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2005.

[12] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, pp. 2454–2466, 2012.

[13] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, 2004.

[14] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[15] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[16] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Deroski, "Hierarchical annotation of medical images," *Pattern Recognition*, vol. 44, pp. 2436–2449, 2011.

[17] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, "The bas-relief ambiguity," *International Journal of Computer Vision (IJCV)*, vol. 35, pp. 33–44, 1999.

[18] S. McCann and D. G. Lowe, "Spatially local coding for object recognition," in *Asian Conference on Computer Vision (ACCV)*, 2012.

[19] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[20] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 2032–2047, 2009.

[21] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, pp. 213–238, 2007.

[22] P. Isola, J. Xiao, A. Torralba, and A. Oliva, "What makes an image memorable?" in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.