Ancient coin classification based on recent trends of deep learning

Sehrish Manzoor, Nouman Ali, Muhammad Raees, Khan Awais Khan, Muhammad Usama Ayub, Afzal Ahmed

Department of Software Engineering, Mirpur University of Science and Technology(MUST), Mirpur- AJK 10250, Pakistan.

Abstract

Automatic ancient coin classification based on pattern recognition is an open research problem. Ancient coins have significant value in cultural heritage and the originality of coins is imperative for their significance. Ancient coin categorization is complex due to the variety in coin designs, purposes, and symbol interpretability thus, drawing a growing amount of research. Here, we present a framework for automatic recognition performed on ancient roman coins. The framework uses a hierarchical knowledge structure of coins complemented with a CNN-based category classifier. We focused on roman republican image datasets for coin identification using CNN Alex-net architecture. Recognition comprises feature-based identification of comparable image matching. Points of interest are detected and described by local characteristics as their appearance. Results are evaluated using the accuracy and precision metrics on Roman Republican Coin Dataset (17546 images) and Roman Coins Dataset (180 Images). Our model outperforms various deep learning methods on this dataset, with a classification ac-curacy of more than 96%.

Keywords

Coin Recognition, Image Classification, CNNs; Alex-Net.

1. Introduction

Ancient coins have utmost importance in cultural heritage engendering vast information about social and political endeavors of a historic era. Ancient coins from a civilization exhibit certain patterns depicting cultural characteristics. In numismatics, the recognition of such patterns has paramount significance, yet it is a daunting activity. Coin patterns pose considerable challenges which are somewhat unadmired in other images because of coin designs and distinct features. Coin quality has been the most domineering concern because ancient coins have deteriorated over the years and most of the features degrade over time. Discrepancies across coins of the same class increase and coins data can be embedded with the noise produced by sharp strokes and contaminants. Coins are multi-faceted in terms of shapes and figurative designs, light, colors, and languages.

The variety in nature of ancient coins poses a substantial challenge even to computer recognizers. For instance, a fundamental problem is a diversity in class labels (many coin types), e.g., mediaeval Portuguese coins and Roman Republic coins constitute over 1500 and 550 classes, respectively [1, 2 & 3]. Deep learning has enabled pattern recognition without the need for hand-crafted features. Deep learning models such as Convolutional Neural Networks (CNNs) are composed of multiple hidden neural layers. Each neural layer receives a signal from the previous layer and provides a signal to the next layer. Each neuron may have an adjustable weight determined by its action. CNN uses image datasets to derive ideal and most useful features weighted by numerous neuron connections and weight adjustments. However, there are significant associated computation and memory requirements. Therefore, a more simplified version such as Alex-Net is a good candidate to be used over other models. Alex-Net delivers high performance on a variety of datasets and makes use of GPU efficiently to perform convolutions and other processing. The contributions of this research study are three-fold as under.

VIPERC2022: 1st International Virtual Conference on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding, 12 September 2022

EMAIL: sehrish.se@must.edu.pk (A. 1); nouman.ali@live.com (A. 2); raees.se@must.edu.pk (A. 3); khanawaiskhn@gmail.com (A. 4); m.usamaayub@gmail.com (A. 5); afzal.se@must.edu.pk (A. 6).
ORCID: 0000-0002-0721-201X (A. 2).

1. Hyper-parameters of Alex-Net architecture is tuned with an adaptive learning rate and sparse-cross-entropy loss function that allow models to train on huge parameters, reduces complexity, and provides better performance.
2. The pre-training of deep learning methods for coin-recognition tasks is performed on unstructured ancient coin images reducing the time for re-training of models from scratch.
3. The results are evaluated on two different datasets of Roman Republicans coins. The competitive analysis of computational complexity and performance shows the Alex-Net model outperforms the state-of-the-art. The approach will also help to examine and set the benchmark for prospective applications in coin classification.

A brief overview of the study is provided in the section. Section 2 outlines an overview of the related work. Datasets and model description is provided in section 3. Section 4 presents results and experimental evaluations. Finally, we conclude our paper in section 5.

2. Related Work

Although coin classification methods for identification and categorization have been researched in recent times, these mostly depend on expensive human involvement. Studies look at the numismatic of pictures by coin categorization, identification, and segmentation [5]. Various methods have been applied to study the domain in detail including neural network models [6], decision trees [3], edge detection [1], gradient directions [7], and contour and texture characteristics [8, 9]. Deep learning, which learns at different abstract levels and has different layers for functions, has the most prominent use above other methods [10]. Excessive studies have been carried out using computer vision methods in coin analysis focused on current coins [11, 12, & 13]. Visual analysis can be carried out using holistically representations like raw appearance [14] or edges [15], as well as off-the-shelf learning approaches like principal component analysis [14] or traditional neural networks [16]. However, in the context of ancient numismatics, such approaches hold little promise. The study in [17] describes, incorporating multiple sources of image information, a method for image-based classification carried out on Roman-Republican ancient coins.
In [18] SIFT descriptors are used to get a 90% accuracy classification on 390-coin images where there have been only 3 classes. CNNs are used to categorize the old Roman Imperial coins on their faces for portrait identification [19]. Coin-Net [20], a new network model proposed a method with feature attention layers. The method also employs pooling known as “compact bilinear” and uses residual groups. They have compiled a diversified image library of Roman Republican coins (one of the largest), which includes over 17546 photos from 228 different reverse themes. An edge-based statistical classification system known as COIN-O-MATIC [21] is built to emphasize dependability and speed for the MUSCLE CIS Coin Competition. Another study [22] try to classify certain old coins based on histogram workmanship and artificial neural networks training, but they have only just performed on 20 coins and their approach delivers only 75% less accuracy than other algorithms in this field. Roman Republican coins were recognized with SIFT support machine capabilities (SVM) [23].

3. Method and Model Design

Research on ancient coin analysis has limited reported evaluation approaches. Limitations of dataset size are evident in coin-image due to their scarcity. Most datasets are also highly controlled and have uniform shapes and features thus restricting generalizability of evaluated methods. We designed our trials to address issues of previous research [29, 30] by using a comparatively large dataset. The research study is carried out by examining two datasets. The first dataset referred to as RRCD is the largest ancient coin image set [24]. The dataset depicts a pattern for most coin images that highlight discriminative features on one side than the other. RRCD consists of around 17546 images of coins with 100 different classes. We also used a smaller dataset to test the generalizability of the model on the small training examples and test examples. The second dataset contains about 180-coin images. A description of random images is listed in Figure 1.

Fig. 1. Sample images from the evaluation database's RRCD 100 classifications (left) [20] from the evaluation database's Roman Coins 60 classifications (right) [17]

We used the Alex network model for multiple datasets to compare the accuracy of the labelled image data used for training. We use a classic comparable Alex-Net structure with five convolutional, three max-pooling, drop-out and three fully connected layers. Convolutional layers use max-out functions, except for the fully connected layers which use a soft-max function. A typical architecture of Alex-Net is depicted in figure 2. The first layer receives the input images using Gray-level photos represented in a two-dimensional input attribute vector. For the RGB, images the input layer has three channels to accept three equivalent inputs in a similar format for red, green, and blue channels, separately. Convolution is a most integral part of CNNs for vision and analysis of pixel values of input images. It involves the multiplication of each set of pixels with pre-defined weight kernels. The kernels learn the prominent and most obvious patterns to define higher-level features by scanning images with sliding windows (filters) which eventually define shapes which predict the image classes. A convolution is a mathematical production of input features X and some function f (weights of the layer), and write the expression:

$$Z = X * f \tag{1}$$

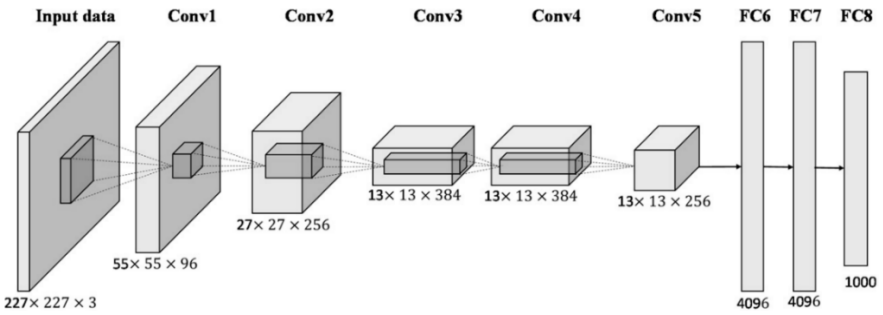

Fig. 2. Architecture of Alex-Net [4]

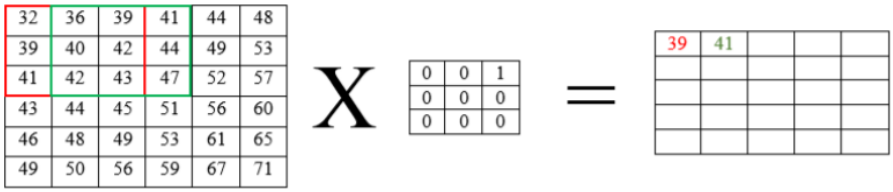

Fig. 3. Convolutional Layer Concept [31]

The detection of image features is done through a kernel or filter. The illustration of performing convolutions by filters (sliding window), is depicted in figure 3. The filter passes through all the values that can be captured by it and output the image pixels multiplied by a weight matrix. The resultant values provide a new matrix or feature map. The process of running a 3*3 filter on a single channel image is shown in the figure. The filter also has a step that defines after each scanning how much it must jump forward (in this case left and down) to capture new values. Similarly, padding can be applied to rinse the boundary values and only focus on the predominant features. 'SAME' padding is utilized to signify that the result is set to the same output image size after convolution as the input image. Principally, padding reduces the loss of data in image corners. The process depicted in the figure refers to the procedure performed for a channel image of 3x3 RGB with 2 sizes and steps=1 filter [31]. Each convolutional layer has a corresponding layer activation function to learn from pixels. We use the Rectified Logic

Unit (RELU) activation function which is a non-linear operation at the end of each convolutional process. It sets all the negative picture pixels to zero on the feature maps. Therefore, any dark (unimportant) features will be marked with zeros resulting in only relevant features being taken to the next layers. The following is the function that performs this normalization:

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k+\propto \sum_{j=max\left(0,i-\frac{n}{2}\right)}^{min(N-1,i+\frac{n}{2})}\left(a_{x,y}^{j}\right)^{2}\right)\beta} \quad (2)$$

Where $a^{i}_{x,y}$ represents the activity of a neuron at a specific layer coordinate (x, y) from kernel i, $b^{i}_{x,y}$ is the corresponding output, n is the number of adjacent kernel maps, N is the total number of kernels in the layer, and the other variables are learned hyper parameters determined using a validation set. RELU is motivated to foster nonlinearity in the CNN, while more and more true information is not linear for our CNN to learn [27]. RELU also performs other functions such as down sampling or spatial pooling by reducing the resolution of the image and considering only the most prominent details inside images. RELU has been used widely and recommended as a useful activation function [27] in comparison with tanh and sigmoid functions. RELU output method and its visual representations are shown in figure 4.

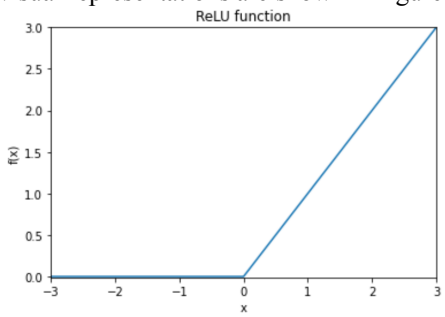

Fig. 4. Graphical representation of Relu Function

Another important element in the CNNs to introduce non-linearity, reduce dimensionality, and minimize calculations is referred to as pooling. Pooling is responsible to retain important elements to the output vector by summarizing section maps. We employed max pooling without any padding on an image which is the most common type of pooling alongside average pooling. A typical description o max-pooling is depicted in figure 5. Max-pooling is controlled by a hyperparameter-defined filter that takes the maximum of matrix values and ignores the rest.

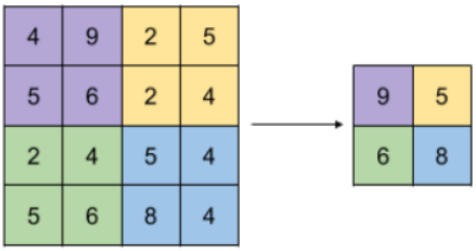

Fig. 5. Max Pooling illustration

After the convolutions and pooling, a flattening layer is added to flatten the features discovered by the previous layers depending upon the dimensions of the input images. Finally, there are fully connected (FC) layers applying the soft-max classification function. Here, the architecture has the flexibility to include other classifiers such as other functions or models such as SVM. The outcome is derived from the fully connected layers after the convolution and pooling layers. Alex-Net is one of the most efficacious adaptations of CNN architectures and we used the conventional network protocol to implement it on the coin-image datasets. It employs 11x11, 5x5, and 3x3 filters, dropouts, pooling, RELU activations, and stochastic optimization implementations. RELU is attached after each convoluted layer to identify only prominent features. Alex-Net searches the image squares for efficiency within the image database. Experimental settings are defined with the input image size 227x227, momentum 0.9, solver test iteration 10, various batch and epochs, an adaptive learning rate set through “ADAM”, a dropout of 0.5, maximum number of iterations 160 (160 epochs) for 17546 images. We used optimal Alex-net parameters apart from the last fully connected layer, where the number of output neurons is equal to the number of classes

(100) in our dataset. We employed the following mechanism to perform the classification task on the ancient coin dataset.
Step 01: **Input**: We have used 17546 color images of size 227x227x3.
Step 02: **Conv-1**: The first convolutional layer consists of 96 kernels of size 11×11 applied with a stride of 4 and padding of 0 and output is calculated by expression:

$$O = \frac{(n-f+2P)}{S+1} \tag{3}$$

Step 03: **MaxPool-1**: The max pool layer following Conv-1 consists of a pooling size of 3×3 and stride 2.

$$h_{xy} = \max_{i=0\ldots s, j=0\ldots s} (x+i)(j+y) \tag{4}$$

Step 04: **Conv-2**: The second conv layer consists of 256 kernels of size 5×5 applied with a stride of 1 and padding of 2.
Step 05: **MaxPool-2**: The max pool layer following Conv-2 consists of a pooling size of 3×3 and a stride of 2.
Step 06: **Conv-3**: The third Conv layer consists of 384 kernels of size 3×3 applied with a stride of 1 and padding of 1.
Step 07: Conv-4: The fourth Conv layer has the same structure as the third conv layer. It consists of 384 kernels of size 3×3 applied with a stride of 1 and padding of 1.
Step 08: **Conv-5**: The fifth Conv layer consists of 256 kernels of size 3×3 applied with a stride of 1 and padding of 1.
Step 09: **MaxPool-3**: The MaxPool layer following Conv-5 consists of a pooling size of 3×3 and a stride of 2.
Step 10: **Fully Connected-6**: The first fully connected layer has 4096 neurons.

$$z_l = W_l * h_{l-1} \tag{5}$$

Step 11: **Fully Connected-7**: The second fully connected layer has 4096 neurons.
Step 12: **Fully Connected-8**: The third fully connected layer has 1000 neurons.

$$z_i = \frac{e^{zj}}{\Sigma_j\, e^{zj}} \tag{6}$$

The complete flow of the process is shown in figure

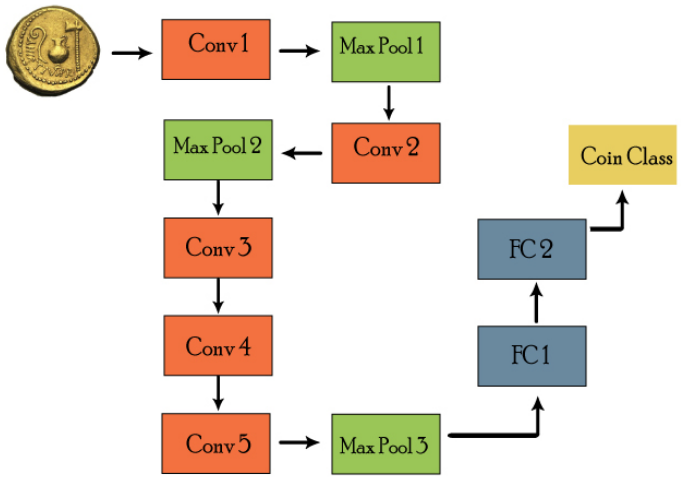

Fig. 6. Flow of Model [32]

4. RESULTS

We used the Tensor Flow and KERAS for the implementation of Alex-Net on the coin-dataset. We utilized the "Cv2" library for processing the images in efficient calculation using NumPy. We split the dataset into two halves - train set, and test set: 90% and 10% of images are used for training and testing respectively. Due to diversity of coins dataset more training images are needed, and less testing images will suffice to understand coin patterns. This architecture uses a sequential model in which each layer is connected to its adjacent layer ahead and back.

The learning rate, which is adaptive, is controlled by the optimizer. The "ADAM" optimizer is utilized for the architecture which modifies the learning rate during the learning phase for optimal performance. Although a lower learning rate may achieve optimum weights, it does not suffice to provide optimal training time. Hence, a reciprocal relationship exists between learning rate and learning time. Secondly, the loss function measures the deviation of the prediction from the actual classes of image data. Here, the 'sparse_categorical_crossentropy' method is used for loss calculation. The composite implementation and working of the model in Tensor Flow are described in table 1. Various attributes are reported as parameters in Alex-Net consisting of five orthodox convolutional layers, and three layers of each max-pooling and full connectivity, respectively. Overall, it comes out to a whopping 28,964,213.

Table 1. The description of working of each layer in Alex-Net architecture

Layer #	Type of Layer	Input	Output	Function	Trainable Parameters
1	Convolution Layer	224×224×3) image	(55×55×96) tensor	RELU activation	34,944
2	Max Pooling	(55×55×96) tensor	(27×27×96) tensor	Max-Pooling	0
3	Convolution Layer	(27×27×96) tensor	(27×27×256) tensor	RELU activation	614,656
4	Max Pooling	(27×27×256) tensor	(13×13×256) tensor	RELU activation	0
5	Convolution Layer	(13×13×256) tensor	(13×13×384) tensor	RELU activation	885,120
6	Convolution Layer	(13×13×384) tensor	(13×13×384) tensor	RELU activation	1,327,488
7	Convolution Layer	(13×13×384) tensor	(13×13×256) tensor	RELU activation	884,992
8	Max Pooling	(13×13×256) tensor	(6×6×256) tensor	RELU activation	0
9	Fully Connected	(6×6×256) tensor	(4096×1) tensor	RELU activation	4,198,400
10	Fully Connected	(4096×1) tensor	(4096×1) tensor	RELU activation	167,81,312
11	Fully Connected	(4096×1) tensor	(1000×1) tensor	soft-max activation	4,097,000

The comparison tables depicted below show an analysis of our approach with the other studies. The evaluation of studies which used the same datasets is performed to compare the accuracy. Table 2 shows a comparison of our approach with the SIFT (Coarse-to Fine) executed on the Roman Coins (180 images). Our approach outperforms the SIFT approach considerably. Our approach reports 100% of accuracy as compared to 83.3% using SIFT (coarse-to Fine). Likewise, Table 2 also shows a comparison of the proposed method two approaches i.e., Bag of Visual Words (BOVW) and Rectangular Tiling (RT). Again, the performance criterion is evaluated on the same dataset of the Roman Republican Coin Dataset (RRCD) of 17546 images. Our method again performed significantly better than the other candidate approaches (BOVW: 71% accuracy, RT: 84% accuracy) by achieving over 96% accuracy.

The performance of the model is evaluated by accuracy, and we got an accuracy of 100% on a set of 60 classes of Roman coins comprised of 180 images (each class is represented by three-coin images of the reverse side. Figure 7 shows the performance of the model for experiment 1 which was performed on dataset 2 (Roman Coins). The image set was simple, and we were able to achieve 100% accuracy after the initial training. Similarly, the training loss drops to zero at a considerable speed. However, training loss generally rises at the start but then stabilizes at an acceptable level of under 2. Experiment 1 is run using 80 epoch and get an accuracy of 100% on dataset 2. The number of epochs has an impact on accuracy, so experiment 1 is carried out to assess the impact of decreasing the number of epochs while maintaining the same number of images in dataset 2, here we gradually decreased the number of epochs to 80, 50 and 10 and achieve accuracy of 100, 70 and 37 respectively.

Table 2. Comparison of Alex-Net with other Models for Roman Republican Coin RRCD (17546) and Roman Coins Dataset (180 images)

Model	Accuracy
Roman Republican Coin Dataset RRCD	
Alex Net (Proposed)	96.3%
BOVW [20]	70.8%
RT [20]	84.4%
Roman Coins Dataset	
Alex Net (Proposed)	100%
SIFT (Coarse-to Fine) [17]	83.3%

Table 3. Experiments and Results

Exp#	No. of Epochs	Accuracy	Images
	Roman Coins Data Set		
Exp#1 E1	80	1.0	180
Exp#1 E2	50	0.7	180
Exp#1 E3	10	0.37	180
	Roman Republican Coin Dataset RRCD		
Exp#2. E1	80	0.94	17546
Exp#2. E2	160	0.96	17546
Exp#2. E3	120	0.95	17546
Exp#2. E4	110	0.952	17546
Exp#2. E5	190	0.963	17546

Roman Republican coin dataset is used for experiment 2, which contains approximately 17546-coin images. The more epochs we run, the more the model is improving, up to a certain point. After that point, the model will stop improving during each epoch. In addition, the more epochs, the longer the model will take to run. For experiment 2 we achieved 96.3% accuracy on the Roman Republican coin dataset that contains 17546-coin images of 100 classes and the number of epochs is the same as for experiment 1. Several epochs have an impact on accuracy, so experiment 2 is carried out to assess the impact of increasing the number of epochs while maintaining the same number of images in dataset, here we gradually increased the number of epochs to 100, 110, 160 and 190 and achieve accuracy of 95.3, 95.2, 96.5 and 96.3 respectively. Figure 8 shows the measure loss and accuracy for the experiment 2. Results of both experiments are shown in table 3. The training loss generally goes down with initial epochs and steadies itself after some iterations. There is an increase in accuracy while increasing the epochs from 80 to 160 gradually and slowly. The accuracy increases from 94% at 80 epochs to 96% at 160 epochs.

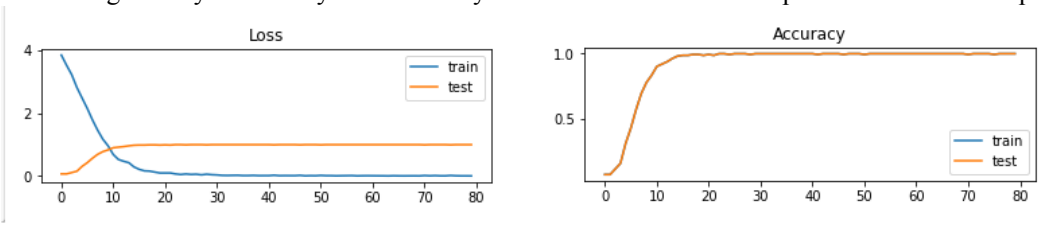

Fig. 7. Loss and accuracy measures during training and testing on Roman Coins dataset 2.

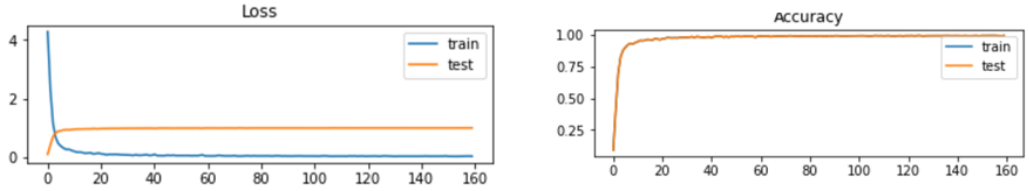

Fig. 8. Loss and accuracy measures during training and testing on RRCD.

5. CONCLUSION & FUTURE WORK

Pattern recognition in ancient coin classification to enhance cultural understanding has open research directions. This study evaluates the working of Alex-Net on multiple datasets using different parameters. Various trials were conducted to obtain a higher accuracy. Several research works have been done on these tasks to boost the performance of CNN. This paper not only describes the working mechanism of CNN's usage but evaluates a modified deep CNN model (Alex-Net) for the detection and identification of ancient coin images. This topic has been examined in a few papers, but no standard answer has been adopted by other researchers. The results of the best model showed that coin images are needed in the training stage, but that all the varied sizes of the coin should be evaluated. Our results show that a large, deep convolutional neural network can achieve record-breaking results on a highly challenging dataset using purely supervised learning. The presented approach can be enhanced by deploying advanced algorithms in real-time in embedded devices.

6. References

1. Bremananth, R., Balaji, B., Sankari, M., & Chitra, A. (2005). A new approach to coin recognition using neural pattern analysis. In Proceedings of INDICON 2005: An International Conference of IEEE India Council (Vol. 2005, pp. 366–370). https://doi.org/10.1109/INDCON.2005.1590191
2. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2015)
3. Aslan, S., Vascon, S., & Pelillo, M. (2020). Two sides of the same coin: Improved ancient coin classification using Graph Transduction Games. Pattern Recognition Letters, 131, 158–165. https://doi.org/10.1016/j.patrec.2019.12.007
4. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks (AlexNet)"
5. Akyuz, T., Mukhamedshina, N., Basaran, S., Akyuz, S., Mirsagatova, A. A., Bolcal, C., & Gulec, A. (2007). Neutron activation analysis of the metals from ancient coins dating back to roman age. Asian Journal of Chemistry, 19(3), 1832–1836.
6. Anwar, H., Zambanini, S., Kampel, M., & Vondrovec, K. (2015). Ancient coin classification using reverse motif recognition: Image-based classification of roman republican coins. IEEE Signal Processing Magazine, 32(4), 64–74. https://doi.org/10.1109/MSP.2015.2409331
7. Cooper, J., & Arandjelović, O. (2020). Learning to Describe: A New Approach to Computer Vision-Based Ancient Coin Analysis. Sci, 2(2), 27. https://doi.org/10.3390/sci2020027
8. Cooper, J., & Arandjelović, O. (2020). Understanding Ancient Coin Images (pp. 330–340). https://doi.org/10.1007/978-3-030-16841-4_34
9. Elkins, N. T. (2008). A Survey of the Material and Intellectual Consequences of Trading in Undocumented Ancient Coins: a Case Study on the North American Trade. Frankfurter Elektronische Rundschau Zur Altertumskunde, 7, 1–13. Retrieved from http://law-archaeology.gr/ClientFiles/Articles/ElkinsFERA.pdf
10. Fare, C., & Arandjelović, O. (2017). Ancient roman coin retrieval: A systematic examination of the effects of coin grade. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10193 LNCS, pp. 410–423). Springer Verlag. https://doi.org/10.1007/978-3-319-56608-5_32
11. Haridas, R., & R L, J. (2019). Convolutional Neural Networks: A Comprehensive Survey. International Journal of Applied Engineering Research, 14(3), 780. https://doi.org/10.37622/ijaer/14.3.2019.780-789
12. P. Davidsson. Coin classification using a novel technique for learning characteristic decision trees by controlling the degree of generalization. In Proc. International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pages 403–412, 1996.
13. M. Nolle, H. Penz, M. Rubik, K. Mayer, I. Holl ¨ ander, and ¨ R. Granec. Dagobert - a new coin recognition and sorting system. In Proc. Internation Conference on Digital Image Computing, 2003.
14. X. Pan and L. Tougne. Topology-based character recognition method for coin date detection. In Proc. IEEE International Conference on Image Analysis and Processing, 2016.
15. R. Huber, H. Ramoser, K. Mayer, H. Penz, and M. Rubik. Classification of coins using an eigenspace approach. Pattern Recognition Letters, 26(1):61–75, 2005.
16. L. van der Maaten and P. Boon. COIN-O-MATIC: A fast system for reliable coin classification. In Proc. MUSCLE CIS Coin Recognition Competition Workshop, pages 7–18, 2006.
17. Zambanini, S., & Kampel, M. (2013). Coarse-to-fine correspondence search for classifying ancient coins. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 7729 LNCS, pp. 25–36). https://doi.org/10.1007/978-3-642-37484-5_3
18. Kampel, M., Zambanini, S., Schlapke, M., & Breuckmann, B. (2009). Highly detailed 3D scanning of ancient coins. In Proc. of the XXII CIPA Conf. Kyoto, Japan.
19. Schlag, I., & Arandjelovic, O. (2017). Ancient Roman Coin Recognition in the Wild Using Deep Learning Based Recognition of Artistically Depicted Face Profiles. In Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017 (Vol. 2018-January, pp. 2898–2906). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICCVW.2017.342
20. 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
21. L. Van Der Maaten and E. Postma, Towards automatic coin classification. na, 2006. 191
22. IqbalQuraishi, Md., Das, G., Gopal Dhal, K., & Das, P. (2013). Classification of Ancient Coin using Artificial Neural Network. International Journal of Computer Applications, 62(18), 6–9. https://doi.org/10.5120/10178-4943
23. Zambanini, S., Kavelar, A., & Kampel, M. (2014). Classifying ancient coins by local feature matching and pairwise geometric consistency evaluation. In Proceedings - International Conference on Pattern Recognition (pp. 3032–3037). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICPR.2014.523
24. Tang, J., Yuan, F., Shen, X., Wang, Z., Rao, M., He, Y., … Wu, H. (2019, December 1). Bridging Biological and Artificial Neural Networks with Emerging Neuromorphic Devices: Fundamentals, Progress, and Challenges. Advanced Materials. Wiley-VCH Verlag. https://doi.org/10.1002/adma.201902761

25. Kim, J., & Pavlovic, V. (2015). Improving ancient roman coin recognition with alignment and spatial encoding. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 8925, pp. 149–164). Springer Verlag. https://doi.org/10.1007/978-3-319-16178-5_10
26. Jorda, M., Valero-Lara, P., & Pena, A. J. (2019). Performance Evaluation of cuDNN Convolution Algorithms on NVIDIA Volta GPUs. IEEE Access, 7, 70461–70473. https://doi.org/10.1109/ACCESS.2019.2918851
27. Khashman A., Sekeroglu B. and Dimililer K., "Intelligent Coin Identification System", Proceedings of the IEEE International Symposium on Intelligent Control (ISIC'06), Munich, Germany, 4-6 October 2006, pp. 1226-1230
28. Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. International Journal of Engineering Applied Sciences and Technology, 04(12), 310–316. https://doi.org/10.33564/ijeast.2020.v04i12.054
29. Shang, W., Sohn, K., Almeida, D., & Lee, H. (2016). Understanding and improving convolutional neural networks via concatenated rectified linear units. In 33rd International Conference on Machine Learning, ICML 2016 (Vol. 5, pp. 3276–3284). International Machine Learning Society (IMLS)
30. Zambanini, S., & Kampel, M. (2011). Automatic Coin Classification by Image Matching. VAST11: The 12th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, (January), 65–72.
31. Jorda, M., Valero-Lara, P., & Pena, A. J. (2019). Performance Evaluation of cuDNN Convolution Algorithms on NVIDIA Volta GPUs. IEEE Access, 7, 70461–70473. https://doi.org/10.1109/ACCESS.2019.2918851
32. Sunny, M. S. H., Dipta, D. R., Hossain, S., Faruque, H. M. R., & Hossain, E. (2019, May). Design of a convolutional neural network-based smart waste disposal system. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5). IEEE.