Automatic Coin Classification

Ştefan N. Tică, Costin A. Boiangiu and Andrei Tigora

***Abstract*—**An automatic system which classifies coins is presented and discussed. The system is flexible, being able to identify coins with various appearances and photographed in different light conditions. For this purpose, a set of robust techniques for thresholding, edge detection and frequency transform were employed in order to generate a fingerprint as significant as possible and as invariant as possible for every coin class.

***Keywords*—**Canny edge detection, coin classification, DCT transform, lighting invariant, Otsu thresholding, rotation invariant.

I. Introduction

A system for automatic coin classification can be used by institutions working with historical coins, thus helping historians in their work. It can also be a useful tool for private coin collectors and other enthusiasts. In this sense, a free online service for coin classification and recognition can be put to work for the community. Another possible application field is the sorting and classification of the large amounts of European coins that were collected after the introduction of the euro.

II. Related Work

Several coin recognition approaches are mentioned in the literature.

Fukumi et al [1] describe a system based on a rotation-invariant neural network that is capable of distinguishing Japanese coins (a 500 yen and a 500 won piece). Rotational invariance is achieved by explicitly generating the rotational group for a coarse model of the coin in a preprocessing step and feeding the results into a neural network. One drawback of the neural network approach is that it is not apparently clear how rejection of coins should be expressed. It is essential to be able to reject coins as it is impossible to know in advance which types of coins will be feed to the system.

Davidsson et al [2], [3] compare several strategies, namely induction of decision trees [4], neural networks and Bayesian classifiers. They derive a variant of the decision tree algorithm that will reject coins if their defining attributes are outside an acceptance margin. However, it is difficult to extend the approach to images.

Adameck et al [5] presented an interesting method for a coin recognition system based on color images. Translational invariance is achieved through segmentation, whereas rotational invariance is a result of a polar coordinate representation and correlation.

III. Feature Extraction

Coin classification performances (accuracy in similar coins identification) depend on the features extracted from coin images. There are three main properties (features) related to coin images [6]:

- the picture on the coin (stamp)
- the texture on the coin
- the text on the coin

The current approach focuses on the first two properties, as described in the following.

A. Stamp (picture) representation

This is the most important part of a coin's fingerprint, based on the statistical distribution of the edges on the coin's surface. Edge feature extraction is a process consisting of three steps:

- edge detection
- edge distance distributions
- edge angle distribution

A similar approach describing the edge distance distributions and edge angle distributions is presented in [7].

1 Edge detection

Based on Rangarajan's presentation of a number of different edge detection algorithms [8] and on our practical tryouts, the conclusion is that the most suitable one for the current project is Canny's edge detection algorithm. Experiments were also carried with the Sobel kernel edge detection, Prewitt's operator, Robert's cross operator and some another method based on an authors idea which turned out to produce results similar with the Sobel kernel approach at the same time being more computational intensive. There are however other potential algorithms, aimed at filtering and improving general edge detectors, that could improve this edge detection [9].

$$\frac{1}{115}\begin{array}{|c|c|c|c|c|}\hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline \end{array}$$

Fig. 1 Discrete approximation to Gaussian function with σ=1.4

Canny’s edge detection algorithm consists in the following six steps [10]:

Step 1. Filter out any noise in the original image. The aim is to prevent detection of false edges. This is done by applying a Gaussian filter or other filters for specific purposes [11]. The Gaussian used in the implementation is presented in Fig. 1.

Step 2. Determine the edge strength by computing the gradient of the image. This is done by first applying the Sobel operator on the image to compute a 2-D spatial gradient on both axes, and then approximating the absolute gradient magnitude (edge strength) for each point. The magnitude is approximated using the following formula:

$$|G| = |G_x| + |G_y| \tag{1}$$

where G_x and G_y are axis oriented gradients obtained by applying two 3x3 convolution masks:

Step 3. The direction of the edge is computed using the gradient in the X and Y directions:

$$\theta = \arctan(G_y / G_x) \tag{2}$$

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Fig. 2 G_x and G_y

Step 4. With the edge direction computed, the next step is to relate the edge to a direction that can be traced in an image. Given a 5x5 image as the one described in Fig. 3, it can be seen, by analyzing pixel "P", that there are only four possible directions when describing the surrounding pixels - 0 degrees (for the horizontal direction), 45 degrees (along the main diagonal), 90 degrees (for the vertical direction), or 135 degrees (along the secondary diagonal).

Step 5. Having estimated the edge directions, non-maximum suppression is applied along the edge and removes any border pixel, resulting in a thin line in the output image.

x x x x x
x x x x x
x x P x x
x x x x x
x x x x x

Fig. 3 Finding Possible directions

Step 6. Finally, a hysteresis processing is employing as a means of eliminating potential streaking. This outcome, which is the breaking of an edge contour, might occur due to the operator output fluctuating above and below the threshold. Hysteresis uses a high and a low threshold instead of a single one. A pixel whose value is greater than high threshold T1 is presumed to be an edge pixel, and is marked as such immediately. Any pixel connected to this edge pixel whose value is greater than the lower threshold T2 is also selected as an edge pixel. Following an edge can therefore be expressed as finding a gradient of at least T1 and continuing as long as the value does not fall below T2. The method can be improved like in [12].

Otsu’s method [13] is a widely used binarization algorithm, which has spawned numerous variants and adaptations [14], [15], due to its robustness and high speed. It is a threshold based global binarization algorithm that aims to maximize the inter-class variance (or minimize the intra-class variance). The inter-class variance σ_b^2 is defined as follows:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{3}$$

where $\omega_x(t)$ represents the class probability and $\mu_x(t)$ represents the class mean. These two are given by the following formulas:

$$\omega_x(t) = \sum\nolimits_{i=0}^{t} c(i) \tag{4}$$

$$\mu_x(t) = \sum\nolimits_{i=0}^{t} c(i) \cdot i \tag{5}$$

with $c(i)$ representing the number of pixels in the image whose value is i. Intuitively the equation tries to separate the means of the cluster, while also keeping each cluster with a high probability of occurrence.

The aim of Otsu’s method is to find the threshold t^* such that:

$$t^* = \arg\max_{t} \sigma_b^2(t). \tag{6}$$

An exhaustive search is employed. For multiple optimal thresholds the value taken into consideration is the mean value [16]. This is very important for the quality of the binarization after the post-processing thresholds stage.

Canny’s edge detector uses two thresholds in its final step. For a given image one can experimentally determine the best values for those two thresholds but this approach is not to be considered in our case as we need a method for setting these values automatically.

The solution to this problem is determining the Otsu’s threshold for a given coin image and derive our thresholds (T1 and T2) based on this value. It was experimentally deducted T1 being equal to 50% of Otsu’s threshold and T2 with 30% of Otsu’s threshold.

In the performed experiments the following problem regarding medieval and older coins classification occurred: those kind of coins usually have an unregulated, bigger stamp height opposed to modern coins which tend to be flattened. Depending on the light conditions and how the shadows are formed when taking the coin's picture, it's possible to have different pictures of the same coin with totally different results regarding the edges detected by Canny's edge detector.

The presented work of coin classification and identification considers photographs of coins taken in an uncontrolled environment, so it is necessary to also overcome the illumination problem. A method that is proposed is based on transforming the image from the RGB space to LUV space [17]. In Fig. 4 is presented a modern coin image and a medieval one transformed in LUV space and its three composing parts: The L, U and V image components were obtained by scaling the LUV values to 0-255.

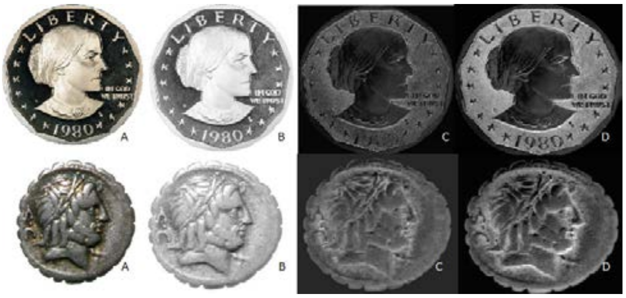


Fig. 4 A modern (upper row) and medieval (lower row) coin picture a) original picture; b) L-component; c) U-component; d) V-component

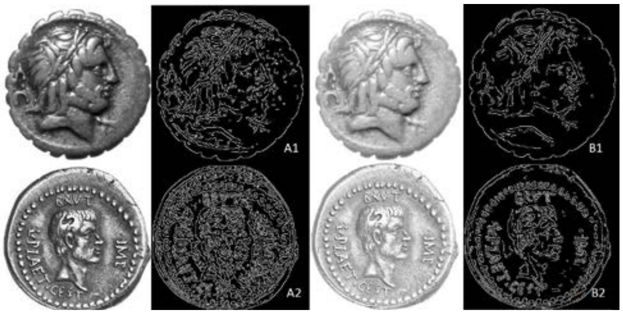


Fig. 5 Grayscale image and corresponding Canny's edges (A1, A2) Image's L-component and associated Canny's edges (B1, B2)

By applying the Canny's edge detector on the image L-component we obtained significantly better results than using the gray scale representation of the image (Fig. 5.).

2 Edge distance distribution

Edge distance distributions measure the distribution of the distances between the pixels on the edges and the center of the coin. The distribution is estimated by dividing the coins in circular concentric parts, as illustrated in Fig. 6. The number of edge pixels in each part is accumulated, and the resulting histogram is normalized in order to provide an estimation of the edge distance distribution. Edge distance distributions are rotation invariant by definition. Edge distance distributions are used in a multiscale approach, by measuring the histograms for various numbers of bins (for 2, 4, 8, 16, and 32 bins).

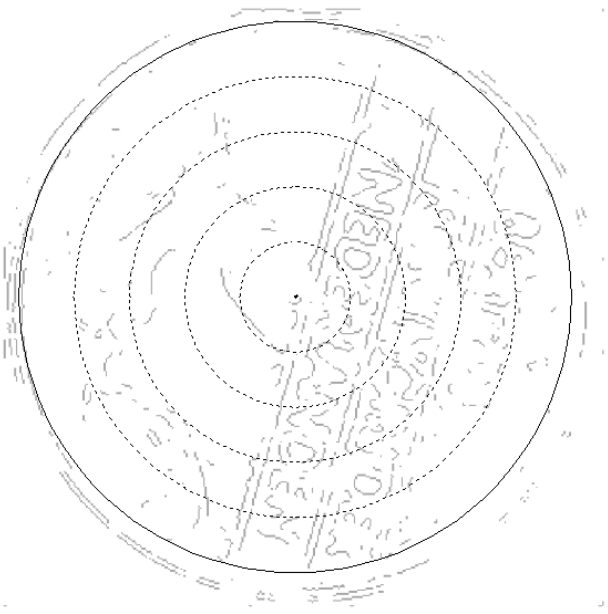


Fig. 6 Distance distribution

3 Edge angle distribution

Although edge distance distributions seem to be strong instruments in coin classification, they do not incorporate all the information. In edge distance distributions, the relative angular distribution of the edge pixels is not represented. The relative angular distribution of edge pixels can be described using edge angle distributions. Edge angle distributions are measured by dividing the coin in pie-shaped parts, as is illustrated in Fig 7. The number of edge pixels in the parts is accumulated, and the resulting histogram is normalized in order to provide an estimation of the relative angular distribution of the coin edge pixels.

In contrary to edge distance distributions, edge angle distributions are not rotation invariant by definition. Rotation invariance of the edge angle feature can be obtained by computing the magnitude of the Fourier transform of the obtained histogram. This step makes the histogram invariant under circular shifts (which correspond to rotations of the coin). In this respect, a large number of bins in the histogram are required, since a rotation of the coin should imply a circular shift on the histogram, instead of a change in the histogram accumulators.

In the current implementation the edge angle histogram is measured in scales of 240 bins compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-order ones.

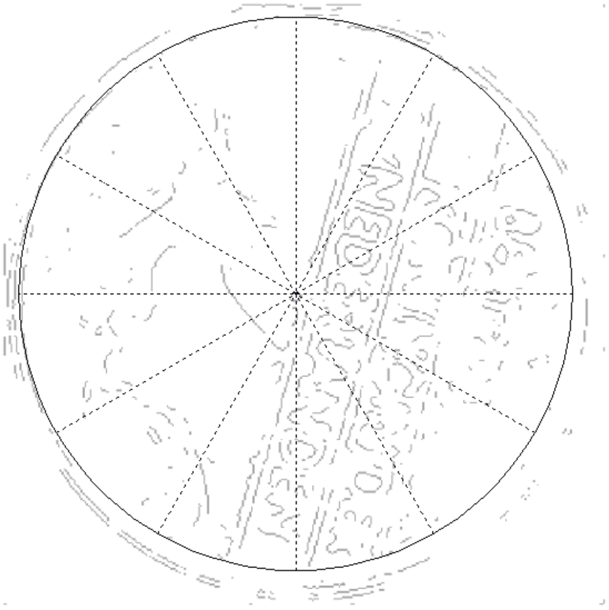


Fig. 7 Angle distribution

B. Texture of the coin

The texture component of the fingerprint is based on two DCT (Discrete Cosine Transform) parts of the LUV representation of the coin image inspired by Menon et al. [18].

The discrete cosine transform helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT transforms a signal from the spatial domain into the frequency domain. The order of values obtained by applying the DCT is coincidentally from the lowest to highest frequency. This feature and the psychological observation that the human eye is less sensitive to recognizing the higher-order frequencies leads to the possibility of compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-order ones.

A 2D-DCT is calculated [19] over the L-channel (Luminosity) of the image. The first coefficient stands for the DC value, or average luminosity of the image. The next coefficients represent the higher order values with increasing frequency. A number of these coefficients (the first 10 coefficients) are taken and normalized for the grayscale fingerprint part. This part of the fingerprints represents the basic composition of the image.

Then a 2D-DCT of the two color components are calculated and used for the color part of the fingerprint. Here only the first three of each are considered, since the human eye is much more sensitive to luminosity than to color. This part of the fingerprint represents the color composition of the image, with reduced spatial resolution compared to the gray scale part.

The 2D DCT is given by the formula:

$$X_{k1,k2} = \sum_{n_1}^{N_1-1}\sum_{n_2}^{N_2-1} x_{n1,n2} \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cdot \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]. \tag{7}$$

IV. Classification and Identification

The following two subsections each deal with one of the topics of classification and identification.

A. Classification

In the previous section, we presented a number of coin-specific features which can be used to label a coin image. This section describes how the features are used in order to decide reliable upon the class of the coin.

The fingerprint of an image is composed of:

- five distance distribution vectors (for 2, 4, 8, 16 and 32 bins)
- a Fourier transformation magnitude vector over the angle distribution for 240 bins
- first 8 components of the DCT over the L-channel
- first 4 DCT components for the U-channel
- first 4 DCT components for the V-channel.

Checking the similarity is done by calculating the nearness between the images by calculating a distance value. For this purpose it’s logical to use the Euclidian distance between two points in a n-space, given by the formula:

$$d(p,q)=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}\,. \tag{8}$$

B. Identification

The identification consists of two phases:

- Firstly, the DCT method for fingerprinting is used to preselect a group of images with similar texture.
- Secondly, the distance and angle distribution method is used to identify the best three matches from the group obtained in the previous step.

V. Results

The experiments were performed on a Core2DUO machine with 2GB of RAM, running 64bit Windows 7.

The test set consisted of 100 different coin pictures, freely offered on the Internet. As so, the coin photos come from different people using different methods for capturing them (in regards of rotation and illumination), as well as varying sizes and dpi.

Determining the features of a coin is a relatively fast process, taking and average of 13.8 milliseconds; however, as mentioned earlier, due to the heterogeneity of the input material, this could go as low as 9 milliseconds, or even as high as 35 milliseconds.

The features of these reference coins were stored in a database, and then used to match against completely different set of test coins. Due to the relatively modest coin collection available, the bulk of the processing was allocated to feature extraction of the target coin as opposed to matching against reference coins. For a real coin catalogue, it is expected that most of the effort would be put into comparing the coin against every single entity in the database, which could be easily parallelizable.

The tests performed on modern coins were quite promising with really good results in respects of accuracy. The incorrect classifications were due only to poor image quality (e.g., a very dark picture).

As expected, tests having as subject medieval or older coins were not as convincing as those having modern ones, but it has to be taken into consideration the fact that there are some problems associated with pre-modern coins [20]: they are not fabricated in a factorial process, the variety in the stamps of medieval coins is inferior to the variety in stamps of modern coins and they are often strongly eroded due to frequent use and passing of time.

VI. Conclusions

The current paper presented a method for automatic classification of coin pictures using two effective feature types, a method which performs very well on modern coins. Classification of medieval coins proved not to be as good as for modern coins. Furthermore, our work showed an approach for dealing with photo capture inconsistencies (e.g. illumination problems).

In the Appendix are presented sample of coins that are usually correctly handled (identified and classified) by the system as well as a set of coins that are frequently misclassified. It can be observed that most of the problems are caused by the advanced state of deterioration of the coins.

VII. Future Work

The future work should be aimed at solving the known misbehaviors of the system but it is fair to mention that a much bigger input database will reduce the current percentage of correct identification and classification.

More specifically, future work will focus on improvements for the case of medieval coins classification. In this respect there will be evaluated a system that uses images of coins and constructs a pyramid of different sizes of the original image.

Each of the system components has a package (a set of unique up-sampling – down-sampling filters) that is used to recursively construct a pyramid.

Generating a new level involves the following operations:

- the image of the current level is down-sampled, obtaining an image half its length and half its width
- the down-sampled image is afterwards up-sampled, obtaining an approximation of the original image
- a residual image is obtained by extracting the differences between the original image and the up-sampled one
- the residual image is used in fingerprinting and classification alongside some of the current fingerprints

Fig. 8 represents an example of the resulting pyramid. At the end of each cycle, the down-sampled image becomes a new layer in the pyramid. The processing continues, generating more levels until a small enough image is obtained. The limit is a linear image, with either only one row or only one column of pixels.

It is possible that this system of pyramidal coding [21] offers better signatures and faster classification (candidates can be rejected from lower pyramidal levels where signatures are easier to compute and to compare one against an entire dataset). Another advantage is that lower levels will be less sensitive to noise, illumination problems and coin degradation.

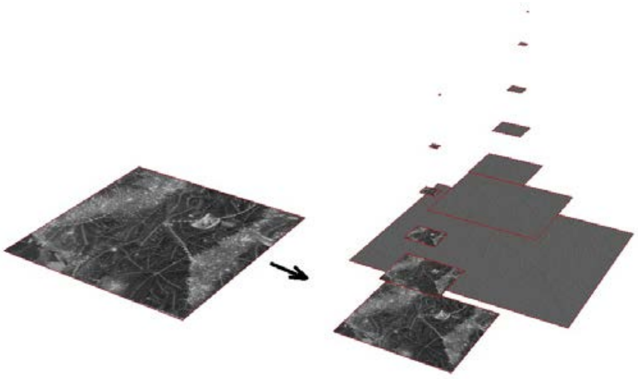


Fig. 8 Constructing a pyramid of different image resolutions

Also, because the direction of looking and position of the source of light is not a factor that should influence which parts of the coins are more visible, a specular removal phase could aid in normalizing the photographic conditions.

When light hits an object, some of the rays will be immediately be reflected by the outer surface, while others will penetrate the object. The rays that managed to get through the surface will either pass through the material (transmitted or absorbed) or will to be reflected out of the object again, but scattered by the material's particles. The immediately reflected rays are called specular reflections and the ones that penetrated the object and were reflected back are called diffuse reflections. Thus, the specular light is highly oriented and not influenced by the material's properties, because it promptly bounced; while the diffuse is scattered by the random particles inside the object and is strongly influenced by them.

In order to be able to separate them more accurately, a good difference between them must be described:

- The reflections have different degrees of polarization (DOP). Specular reflection is generally more polarized than diffuse reflection [22].

- The intensity distribution of diffuse reflections can be well approximated by the Lambert’s Law [23]. The intensity distribution of specular reflections follows the Torrance-Sparrow reflection model [24]. By isolating the diffuse component, powerful Lambertian-based tools can be applied to real world scenes, for recognition and reconstruction.

- The color of specular reflection is determined by the object's surface spectral reflectance, which is mostly constant throughout the visible spectrum. This causes the color of specular reflections to be similar to the light's source, while the color of diffuse reflection to be determined by the object's body spectral reflectance.

We started working on this approach following [25]. Because the specular component resembles the illumination's characteristics, the algorithm starts by normalizing the illumination color using a known or estimated illumination chromaticity (hue and saturation). This will generate an image heaving strictly white specular components. Continuing from this image, an iterative phase shifts the intensity and maximum chromaticity but keeps the hue constant. From this phase results a specular free image, but with altered surface colors. The stop condition is given by a logarithmic differentiation on both the normalized and specular free images, estimating the number of specular pixels left. This process needs only two neighboring pixels, making it local, so segmentation is not needed.

Even though the original authors mention running times of 6 minutes on an Pentium III Computer, first test results for GPU processing, using DirectX 11 and a slightly less precise approach, but with similar results, yield real-time results.

REFERENCES

[1] M. Fukumi, S. Omatu, F. Takeda, T. Kosaka, "Rotation invariant neural pattern recognition system with application to coin recognition," *IEEE Transactions on Neural Networks*, no. 2, 1992, pp. 272–279.

[2] P. Davidsson, "Coin classification using a novel technique for learning characteristic decision trees by controlling the degree of generalization," *Ninth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/ AIE-96)*, 1996, pp. 403–412.

[3] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, 1986, pp. 679-714.

[4] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, no. 1, 1986, pp. 81–106.

[5] M. Adameck, M. Hossfeld, M. Eich, "Three color selective stereo gradient method for fast topography recognition of metallic surfaces," *Machine Vision Applications in Industrial Inspection*, vol. 11, 2003, pp. 128-139.

[6] Ș. N. Tică, C.-A. Boiangiu, A. Tigora, "A method for automatic coin classification," in *Proc. 1st WSEAS International Conference on Image Processing and Pattern Recognition* (IPPR '13), Budapest, Hungary, December 10-12, 2013, pp. 188-198.

[7] L. J. P. van der Maaten, P.J. Boon, "COIN-O-MATIC: a fast and reliable system for coin classification," *Proceedings of the MUSCLE Coin Workshop 2006*, 2006, pp. 7-17.

[8] S. Rangarajan, "Algorithms for edge detection," Available: *http:// www.ee.sunysb.edu/ ~cvl/ ese558/ s2005/ Reports/ Srikanth%20Rangarajan/ submission.doc*, accessed on November 3rd 2012.

[9] M. Zaharescu, I. C. Petrescu, "Edge detection in document analysis," *Journal of Information Systems, Operations Management – JISOM*, May 2013, pp. 156-165.

[10] B. Green, "Canny edge detection tutorial," 2002, Available http : // dasl . mem . drexel . edu / alumni / bGreen / www . pages . drexel . edu / _weg22 / can_tut .html, Accessed on November 23rd 2013.

[11] H. Y. Chai, L. K. Wee, E. Supriyanto, "Edge detection in ultrasound images using speckle reducing anisotropic diffusion in canny edge detector framework," in *Proc. 15th WSEAS international conference on Systems*, 2011, pp. 226-231.

[12] R. Deriche, "Using Canny's criteria to derive an optimal edge detector recursively implemented," *Int. J. Computer Vision*, vol. 1, 1987, pp. 167–187.

[13] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979, pp. 62-66.

[14] S. Bag, P. Bhowmick, P. Behera, G. Harit, "Robust binarization of degraded documents using adaptive-cum-interpolative thresholding in a multi-scale framework," *International Conference on Image Information Processing (ICIIP)*, 2011, pp. 1-6.

[15] L. Jianzhuang, L. Wenqing, T. Yupeng, "Automatic thresholding of gray-level pictures using two-dimension Otsu method," *International Conference on Circuits and Systems. Conference Proceedings*, 1991, pp. 325-327.

[16] A. Tigora, "An image binarization algorithm using watershed-based local thresholding," in *Proc. 1st WSEAS International Conference on Image Processing and Pattern Recognition* (IPPR '13), Budapest, Hungary, December 10-12, 2013, pp.154-160.

[17] D. Bourgin, 2009, Color spaces FAQ, Available: *http:// www.ilkeratalay.com/ colorspacesfaq.php*, accessed on November 1st 2012.

[18] V. Menon, U. Babbar, U. Dasgupta, "Web-based image similarity search," 2009, unpublished.

[19] T.-Y. Sung, "Memory-efficient and high-performance 2-D DCT and IDCT processors based on CORDIC rotation," in *Proc. 7th WSEAS International Conference on Multimedia Systems & Signal Processing*, Hangzhou, China, April 15-17, 2007, pp. 7-12

[20] L. J. P. van der Maaten, E. O. Postma, "Towards automatic coin classification," *Proceedings of the EVA-Vienna 2006*, 2006, pp. 19-26.

[21] W. Zhang, F. Bergholm, "Multi-scale blur estimation and edge type classification for scene analysis," *International Journal of Computer Vision*, vol. 24, no. 3, 1997, pp. 219 – 250.

[22] D. Miyazaki, R.T. Tan, K. Hara, and K. Ikeuchi, "Polarization-Based Inverse Rendering from a Single View," in *Proc. IEEE Int'l Conf. Computer Vision* (ICCV '03), 2003.

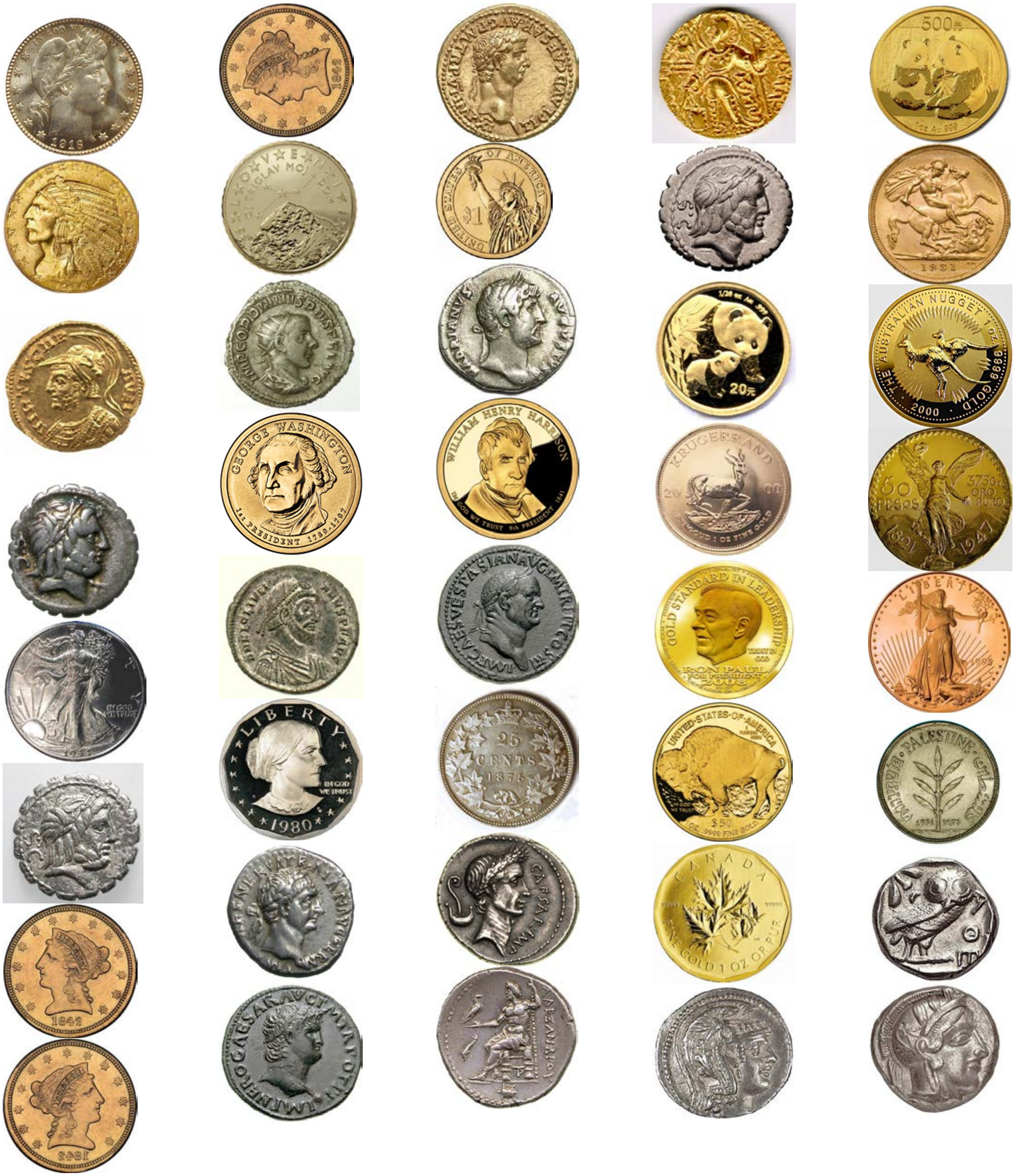
[23] J. H. Lambert, "Photometria sive de mensura de gratibus luminis, colorum et umbrae," Augsberg, Germany, Eberhard Klett, 1760.

[24] K.E. Torrance and E.M. Sparrow, "Theory for Off-Specular Reflection from Roughened Surfaces," *J. Optics Soc. Am.*, vol. 57, 1966, pp. 1105-1114.

[25] R. T. Tan, K. Ikeuchi, "Separating reflection components of textured surfaces using a single image," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, February 2005, vol. 27, no. 2.

APPENDIX A. COLLECTION OF INPUT COINS

(USUALLY CORRECTLY IDENTIFIED AND CLASSIFIED)



Appendix B. Collection of Input Coins

(Sometimes misidentified and/or misclassified)