Universidade do Minho
Escola de Engenharia
Departamento de Informática

Luís Salgado

Medieval coin automatic recognition by computer vision

Reconhecimento automático de moedas medievais usando visão por computador

September 2016

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Luís Salgado

Medieval coin automatic recognition by computer vision

Reconhecimento automático de moedas medievais usando visão por computador

Master dissertation
Master Degree in Engenharia Informática

Dissertation supervised by
Professor Cristina P. Santos

September 2016

ACKNOWLEDGEMENTS

To my parents,

who gave me the curiosity gene.

To my wife,

who patiently puts up with it.

I would like to thank professor Cristina P. Santos for all the support, the precious guidance and the enthusiastic encouragement along this research, without her it would be so much harder to get this far. I am also very grateful to professor Pedro Henriques, who gave me the opportunity to accomplish this project and was always understanding towards my clumsiness in the university protocol.

Some numismatists promptly answered my appeal for categorized images of Portuguese medieval coins and respective permissions of use, which allowed the foundation of an image dataset on this subject. Among these are José Almeida, Paulo Silva, Mário Matos and Alberto Praça.

Finally, I have to both thank and apologize to my wife Clara and my kids for all the time I stole from family time (and, I confess, from household tasks too) to be able to conclude this research. To Clara I owe the revision of this text and its cleansing from the uncountable grammar mistakes.

To all, my enormous gratitude. This work would not be possible without you.

ABSTRACT

Keywords: Computer vision, Image classification, Image segmentation, Machine Learning, Coins

The use of computer vision for identification and recognition of coins is well studied and of renowned interest. However the focus of research has consistently been on modern coins and the used algorithms present quite disappointing results when applied to ancient coins. This discrepancy is explained by the nature of ancient coins that are manually minted, having plenty variances, failures, ripples and centuries of degradation which further deform the characteristic patterns, making their identification a hard task even for humans. Another noteworthy factor in almost all similar studies is the controlled environments and uniform illumination of all images of the datasets. Though it makes sense to focus on the more problematic variables, this is an impossible premise to find outside the researchers' laboratory, therefore a problematic that must be approached.

This dissertation focuses on medieval and ancient coin recognition in uncontrolled "real world" images, thus trying to pave way to the use of vast repositories of coin images all over the internet that could be used to make our algorithms more robust.

The first part of the dissertation proposes a fast and automatic method to segment ancient coins over complex backgrounds using a Histogram Backprojection approach combined with edge detection methods. Results are compared against an automation of GrabCut algorithm. The proposed method achieves a Good or Acceptable rate on 76% of the images, taking an average of 0.29s per image, against 49% in 19.58s for GrabCut. Although this work is oriented to ancient coin segmentation, the method can also be used in other contexts presenting thin objects with uniform colors.

In the second part, several state of the art machine learning algorithms are compared in the search for the most promising approach to classify these challenging coins. The best results are achieved using dense SIFT descriptors organized into Bags of Visual Words, and using Support Vector Machine or Naïve Bayes as machine learning strategies.

RESUMO

Palavras-Chave: Visão por computador, Classificação de Imagens, Segmentação de imagens, *Machine Learning*, Moedas

O uso de visão por computador para identificação e reconhecimento de moedas é bastante estudado e de reconhecido interesse. No entanto o foco da investigação tem sido sistematicamente sobre as moedas modernas e os algoritmos usados apresentam resultados bastante desapontantes quando aplicados a moedas antigas. Esta discrepância é justificada pela natureza das moedas antigas que, sendo cunhadas à mão, apresentam bastantes variações, falhas e séculos de degradação que deformam os padrões característicos, tornando a sua identificação dificil mesmo para o ser humano. Adicionalmente, a quase totalidade dos estudos usa ambientes controlados e iluminação uniformizada entre todas as imagens dos *datasets*. Embora faça sentido focar-se nas variáveis mais problemáticas, esta é uma premissa impossível de encontrar fora do laboratório do investigador e portanto uma problemática que tem que ser estudada.

Esta dissertação foca-se no reconhecimento de moedas medievais e clássicas em imagens não controladas, tentando assim abrir caminho ao uso de vastos repositórios de imagens de moedas disponíveis na internet, que poderiam ser usados para tornar os nossos algoritmos mais robustos. Na primeira parte é proposto um método rápido e automático para segmentar moedas antigas sobre fundos complexos, numa abordagem que envolve *Histogram Backprojection* combinado com deteção de arestas. Os resultados são comparados com uma automação do algoritmo *GrabCut*. O método proposto obtém uma classificação de Bom ou Aceitável em 76% das imagens, demorando uma média de 0.29s por imagem, contra 49% em 19,58s do *GrabCut*. Não obstante o foco em segmentação de moedas antigas, este método pode ser usado noutros contextos que incluam objetos planos de cor uniforme.

Na segunda parte, o estado da arte de *Machine Learning* é testado e comparado em busca da abordagem mais promissora para classificar estas moedas. Os melhores resultados são alcançados usando descritores *dense* SIFT, organizados em Bags of Visual Words e usando Support Vector Machine ou Naive Bayes como estratégias de *machine learning*.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 MOTIVATION AND PROBLEM STATEMENT

The use of computer vision for recognition of coins is well studied and of renowned interest in areas ranging from vending machines to fakes identification. In fact the problem is so relevant that it has been object of financing by the European Union (e.g. Project COINS [1], [2]) and international competitions for researchers (e.g. Muscle, 2006 and 2007). However the focus of research and testing has consistently been on modern coins, under controlled lighting conditions. The focus on ancient and medieval coins could provide valuable tools for archeologists' rapid finding identification; museum curators could be more easily guided on cataloging their collections, or identifying fake or stolen coins; and perhaps historians could find unsuspected relations between different coins from different areas, consequently establishing cultural relations.

The challenge is that the same algorithms that advertise a hit of 97% in recognition of modern coins are quite disappointing when applied to ancient coins.

The discrepancy in these results has several reasons, starting with the nature of ancient coins, that are hand minted, having plenty variances and failures. For example, if we take two pieces of 1 euro (from the same country) they are totally identical and practically indistinguishable from each other, in the case of ancient coins no two are alike copies (if they are equal, surely one of them is a fake). The well-defined standards and uniform thicknesses of modern coins do not exist in ancient coins, which present frequent ripples even within the same figure, not centered or incomplete designs, or irregular shapes. On top of all this are the softer metal alloys and centuries of degradation which further deform the characteristic patterns, making their identification a hard task sometimes even for humans. When we move from the classical coins (like Roman and Greek) to medieval coins, the task gets even harder, due to the severe declination in the artistic quality from artisans in the Middle Ages.

Another noteworthy factor in almost all similar studies is the controlled environments and uniform illumination of all images of the training and testing datasets. This choice makes sense, taking into account the amount of other more problematic variables that must be managed in research algorithms to achieve minimally encouraging results. However this is an impossible premise to find in the real world, where the photographs that populate shops, auction houses, numismatic forums, archeology offices and sometimes even museums are the most disparate imaginable and often with awful lighting conditions. Thus, any methodology that aims to be applied to the world outside the researcher's laboratory will necessarily have to consider this disparity. As this context is the most ill-studied, it took the major focus on the investigation and tests presented on this research.

In an age when image recognition researchers seem to be battling over a few decimal points in success rates, working on medieval coins, especially in such hard conditions, does not foretell to be promising at all, thus, this is not a very appellative area to bet our time and careers on. Nevertheless, the technology is achieving a point that could already start to be used in the real world helping researchers in other areas (like archeologists, museum experts and numismatists) forwarding our global knowledge. This work aims to give a modest contribution to such advance.

1.2 SOLUTION

Dealing with heterogeneous images taken under uncontrolled conditions, the initial and arduous first step is segmenting, i.e. removing the background from the coins' image. Very often the images present highly textured backgrounds or are taken with a coin directly in the hand or in the grounds of an excavation, causing the traditional approaches like Canny Edge or Hough Transform to fail. This research proposes the use of Histogram Backprojection to isolate the area of the coin on the image, using as ground model a section of the same coin's image. The segmentation is then refined using edge detection methods. The result is a black mask that, when added to the original image, will delete its background. This technique demonstrated an improved accuracy over the current state of the art algorithms, at a fraction of the computation time.

A primary approach of this present method was accepted in the 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC 2016) and gave way to a paper to be published by IEEE [3]. Meanwhile the research continued and the method has evolved to include edge information on the detection, as well as further optimizations.

For classification or identification purposes, several approaches were compared. On the mathematical descriptions of the coins, the best results were achieved using dense SIFT descriptors grouped with a Bag of Visual Words strategy. The discrimination of the classes was better solved using Support Vector Machine (SVM), or Naïve Bayes techniques. While Naïve Bayes attained the highest accuracy, SVM proved a more tolerant and robust method as well as being faster.

As an additional result of this work, a labeled dataset of uncontrolled images of medieval coins was collected, along with the respective permissions from the owners for its free use for researching purposes. This dataset will be made available to all researchers, providing an asset for future investigations.

1.3 DISSERTATION OUTLINE

Chapter 2 presents the state of the art with an introduction on general computer recognition and describing the most important work done on coin recognition and relevant studies in other domains.

Chapter 3 details the problems that may be approached using computer vision over ancient and medieval coins and the biggest challenges are explained. It also presents an outline of the premises used in the chosen solution, and describes the testing environment.

In Chapter 4 the first part of the proposed method is presented: the segmentation. The several stages of the algorithm are detailed and the results of the tests are compared to an automatic implementation of the best found alternative method: GrabCut.

The second part of the method is presented in Chapter 5, where the most successful state of the art machine learning algorithms are compared under several variations in order to find the most promising approach to classify ancient and medieval coins.

Final conclusions are presented in Chapter 6.

2

State of the Art

2.1 Image Recognition Evolution

Any attempt to summarize the evolution of image recognition in a couple of pages fails inevitably into over-simplification, nonetheless one may try to describe some of the relevant keystones:

In 1999 Lowe [4] introduced Scale Invariant Feature Transform (SIFT) and marked a rupture with the previously ordinarily used Sum of Squared Distances (SSD) [5]. SIFT allows a point inside an RGB image to be represented robustly by a low dimensional vector and find a way to be invariant to scaling and rotation, partially invariant to illumination changes and robust to local geometric distortion. Many variants and alternatives to SIFT appeared meanwhile, but with images increasingly going online and easily growing into large datasets, more ambitious object recognition problems were arising and these raw descriptors were not enough to deal with deforming objects, occlusions and other new challenges. New methods had to be found. Visual Words (VW) [6] presented a smart way of applying the same principles from text matching to visual content. It can be represented by small parts of an image which carry some kind of information related to the features (such as the color, shape or texture), or changes occurring in the pixels, such as deformations or missing areas. Another popular idea arose: the need to have some sort of binning structure for matching objects. Grids were initially placed around entire images, and later on they would be placed around object bounding boxes. Methods like Pyramid Match Kernel [7] introduced powerful and hierarchical ways of integrating spatial information into the image matching.

The Histogram of Oriented Gradients (HOG) [8] feature descriptor arrived around 2005 and is based on the counting of occurrences of gradient orientation in localized portions of an image, simplifying and greatly improving the processing speed. HOG in combination with a new machine learning tool called Support Vector Machine (SVM) [9] easily gained acceptance. A later technique called the Deformable Parts-based Model (DPM) [10], helped reinforce even more the popularity and strength of the HOG technique.

As datasets became massive, an old and discredit method rose again: Deep Learning [11]. It attempts to model high-level abstractions in data by using multiple processing layers with complex structures, or otherwise composed of multiple non-linear transformations. In the present time Convolutional Neural Networks (CNN) [12] is being tried in virtually every problem and Machine Learning is starting to fuse with Artificial Intelligence.

2.2 COIN RECOGNITION

2.2.1 Recognition of modern coins

One of the first major advances in coin recognition was given by the Dagobert [13] project[1]. Its purpose was to sort high volumes of modern coins. The coins were already singled out and put on a conveyor belt where a camera observed one coin at a time in ideal lighting conditions. The method relied on binarized edge information that was correlated with all possible master edge images stored in a database, finding the master coin with lowest distance. For edge information they used Canny edge operator and Laplacian of Gaussian (LoG), plus a polar coordinate representation, but also sensor information of coin diameter and thickness. The success rate was high but the use of sensors and such very controlled conditions make this method infeasible for the purpose of this work. The massive coin image database was later available to the public, and it still provides the best dataset existent for modern coin recognition.

Huber et al. [14] proposed the use of Eigenspaces in modern coin recognition. The method consisted in a preprocessing, performed to obtain a translationally and rotationally invariant description, followed by a second stage, in which an appropriate Eigenspace was selected.

[1] A presentation video of the Coin Classification Machine is available in www.youtube.com/watch?v=I6JiD2yEi8Q

The MUSCLE CIS Coin Competition[2] in 2006 and 2007 launched new ground in coin recognition investigation. The big winner in 2006 was Reisert et al. [15], who used gradient based orientations in order to achieve the most effective success rate. They segmented the coin from the background by applying the Hough transform, then normalized the region containing the coin and transformed it to polar coordinates. An angular image was then computed based on the image gradient orientations. The similarity between two different coins was computed by counting the number of pixels with which the two respective angles coincide. This similarity measure was fed to a Nearest Neighbor classifier that would find the best-matching coin within a given coin image database. Reisert would later improve the method [16]. The runner-up, called *COIN-O-MATIC* [17], also persists in paper citations. This one focused on reliability and speed, relying on the coin edge information, but also on sensor information. The Edge angle-distance distributions were calculated and classified using the Nearest Neighbor approach. In 2007 Maaten showed up again with a paper over partially occluded coins [18]. For that he used Texton-based texture classifiers and template matching based on gradient orientations. But Zaharieva paper [19] tested a promising new area for coin recognition: Scale-Invariant Feature Transform (SIFT) (originally presented by Lowe [4]), which would prove much more effective in ancient coins.

Other authors experienced new approaches around the same time. Neural Networks were tested by several researchers, like Khashman et al. [20]. Ghanem et al. [21] tried a Gabor wavelet approach for feature extraction followed by Nearest Neighbor classification.

Some later, if more modest, contributions in the modern coin recognition continued the pursuit for higher success rates: Märtens et al. [22], used Cross-correlation (CC) matching (while admittedly CC is not invariant to imaging scale, rotation, illumination and perspective distortions, the authors claim that normalization of the CC can significantly improve the method); Vadivelan et al. [23], who experimented both Gabor wavelet and Local Binary Patterns (LBP) operator features over several distance measurement methods and a Nearest Neighbor classifier; and Wei et al. [24], who presented an approach based on image textures, using Ant colony optimization (ACO) for optimal

[2] Information and data from the competition in: muscle.caa.tuwien.ac.at/coin_past.php

threshold segmentation and Tree-structured Wavelet Transform (TWT) for the Textural Characteristics Extraction.

An interesting survey on Techniques of Coin Detection and Recognition is presented by Mehta et al. in [25]. But, by that time, with all the previous studies and optimistic success rates, the problem of coin recognition seemed solved to most researchers.

Yet, these same methods provided very frustrating success rates when applied to ancient coins, proving that the techniques developed for modern coin classification are not sufficient for ancient coin classification [26], [27], [28]. The reasons were already discussed in the introduction, but some basic assumptions also changed. For instance, the generally accepted use of Hough transforms for segmentation of the image is no longer appropriate because very frequently the coins are not perfectly round. Also, even if one wants to use controlled lighting conditions, regardless of the previous arguments against it, the lighting uniformity is harder to achieve due to the degradation and deformation of the ancient coins. And, of course, the use of sensors does not make much sense when dealing with just a few specimens instead of thousands, or in industrial/commercial applications.

2.2.2 Recognition of ancient coins

Some attention to ancient coin recognition came from the EU sponsored COINS (Combat On-Line Illegal Numismatic Sales) project [1], [2], focusing on fake and stolen coins identification. Its main approach relied on individual, unique features, which make a specimen different from all other individuals in the same class. For over two years, a good number of studies came from this project. The team started with a good analysis of the problem [29], provided a dataset of ancient coins from the collection of Fitzwilliam Museum in Cambridge, and have shown again the potential of SIFT classification. The various problematics were approached in distinct papers. The segmentation problem was discussed in [30] and [31] suggesting the use of local entropy and a local range of grey values, but still facing the big problem of the coin shadows. The image acquisition was also tested in [32], towards an optimized acquisition process. Several books [33], [34], and internet articles already existed on the subject, but the study made sense since these focused on the aesthetics and not on the recognition optimization. Nevertheless the investigation continued to be on controlled images. Some possible numismatic research fields where debated in [35], paving the way for more investigation in the area.

The team presented in [36] an end-to-end coin identification workflow for ancient coins, with decent results. They cross-evaluated the performance of several Interest Point Detectors [Difference-of-Gaussian (DoG), Harris-Laplace, Harris-Affine, Hessian-Laplace, Hessian-Affine, Fast-Hessian, Geometry-based region (GBR), Intensity-based region (IBR), and Maximally Stable Extremal Regions (MSER)] with different local image feature descriptors for coin classification and recognition [SIFT, Gradient Location and Orientation Histogram (GLOH), shape context and Speeded Up Robust Features (SURF)]. Based on the irregular shape of ancient coins, in [26] they introduced the use of a deviation from circular shape matching (DCSM) as a form of identification (not classification), a method they would use in several of their later papers, under the argument that the outline is a unique characteristic of a coin. While this is true and useful when identifying stolen coins, relying on it for the identification of fake coins shows some unawareness on the refined methods used by the fakers. Other problems with the concept are the lighting conditions, which admittedly could influence the results, and the computation time which, according to a newspaper article[3], took "a few minutes".

The fact that the online tools from COINS project come public provided much more feedback about its effectiveness. Some reviews claimed that the results were variable[4], and the segmentation was not impressive, exposing the difficulties of using "real world" photos. By now the internet site along with the online tools have vanished[5], leaving the big public with nothing to work with once again.

Meanwhile a limited number of researchers add their contribution, too. Arandjelović [37] introduced a new concept based on localized analysis rather than taking the coin as a whole. His method is based on a feature he called Locally Biased Directional Histogram (LBDH). For each interest point found by the Difference-of-Gaussian (DoG) detector, a set of weighed and directed histograms is computed. These features aim to capture geometric

---

[3] The article can be read online in diepresse.com/home/techscience/wissenschaft/364117/Keine-zwei-Munzen-gleichen-einander-zu-100-Prozent?_vl_backlink=/home/techscience/index.do.

[4] e.g. digitalhn.blogspot.pt/2009/06/software-from-coins.html.

[5] Although there is still a page about the project in oldwww.prip.tuwien.ac.at/research/completed-projects/coins.

relationships between interest points. This proved a promising kind of feature as it offers a powerful representation able to capture the class-specific coin appearance. The method achieved 52.7% classification accuracy, largely outperforming a histogram of SIFT representation, but mainly it opened the eyes of the community for the need to look at special relations between the interest points.

Allahverdi et al. [38], [39] tested some already known methods on Sassanid coins, which are relevant for this project for being closer to medieval coin's style than the Roman coins used in most investigations. One of the papers explored the Discrete Cosine Transform (DCT), the other used Principal Component Analysis (PCA), plus Bhattacharyya distances between the coefficients vector and those representing each training coin. Both methods presented interesting results but were meanwhile outperformed by later studies. The same kind of coin was later approached by Parsa et al. [40], using a representation of the coin image based on the phase of the 2-D Fourier Transform (FT) of the image so that the adverse effect of illumination was eliminated. Then, a Bi-Directional PCA (BDPCA) approach was used and an entry-wise matrix norm calculated the distance between two feature matrices so as to classify coins.

From 2011 to 2014 the ILAC[6] [41] research project joined again (at least) some of the researchers from COINS project for a new batch of studies. The previous approaches were extended by Huber-Mörk et al. [42] using a preselection step based on the coin's contour. In this step, equally spaced rays are cast from the coin's center of gravity and intersected with its contour. The distances along the rays between these intersection points and the hypothetical perfect circle fitted to the coin area are measured and form a descriptor that can be computed quickly. This descriptor can be quickly matched and allows for fast pruning of large coin databases when attempting to identify a specific coin from an image. The second stage uses preselection by the first stage in order to refine the matching using local descriptors, and the results are combined using naïve Bayesian fusion.

The multi-stage method was also used in classification in [43]. The hierarchical sub-selection scheme showed that the classifier-free classification time could be reduced to one-seventh without a loss of classification accuracy, a crucial gain. This was an

[6] The project site can be found in www.caa.tuwien.ac.at/cvl/project/ilac/.

improvement to a previous paper [44] defending the use of a classifier-free approach, like SIFT flow matching, with the main benefit of making us less dependent on the availability of a large and representative set of training images. This makes total sense when talking about scarce ancient coins. On the other hand the computation times, even using the hierarchical sub-selection scheme, are big and get even bigger as the dataset grows. In their tests, with 180 images, from 60 classes, the average classification time varies between 7 and 472 seconds. With a bigger dataset the times would become too uncomfortable for general public use. The classifier-free and multi-stage viewpoint was maintained later in [45], with computation times around 22s in the same 60-class problem. This time their approach was a data-driven first-order matching and used geometric constraints afterwards to reason about the geometric plausibility of the correspondences found. They also opted out SIFT descriptor in favor of Local Image Descriptor Robust to Illumination Changes (LIDRIC), a descriptor presented in [46] by the same team.

Again Huber-Mörk et al. [47], brought back edge features, in a comparison analysis between two approaches for classification and identification of coins: a method based on matching edge features in polar coordinates representation (as in [13]) and a method for matching based on an Eigenspace representation (as in [14]). Interesting results were achieved for identification with the Eigenspace method using deviation from circular shape matching (DCSM) and SIFT (again, with the limitations already commented on).

Anwar et al. [48] and [49], used Bags Of Visual Words (BoVW) – also called Bags of Features (BoF) – based on densely extracted local features such as SIFT, with spatial information, to propose a new method for classification by recognizing motifs minted on their reverse sides. The dense sampling for BoVW results in a better classification rate as it is capable to capture the underlying geometry of the motif even if some of its parts are missing.

Inevitably, the glamour of Convolutional Neural Networks (CNN) would prove irresistible to be applied in this field too. A good example is the work of Kim et al. [50], concluding that CNN outperforms Support Vector Machine (SVM), but presenting success classification rates no better than the previous papers. There are hundreds of new papers on the subject every year (unfortunately not explicitly on coins) which makes it hard to keep track, but a very good example of what is being made is presented by Szegedy et al. [51].

2.2.3 Unusual approaches for coin recognition

Different approaches have been tried which, even if not applicable in the current study, are very interesting and show that creativity in new methodologies shall not be restrained.

Facial recognition

The fact that a great number of Roman coins display the rulers' bust or some deity inspired a few authors to use some form of adapted facial recognition approaches in coin recognition. Kim et al. [52], [53] approaches the subject with a method based on discriminative Deformable Part Models (DPM).

Character recognition

Most coins have legends, so the use of character recognition to classify a coin or at least to help in the process seems a logical next step. Arandjelović [54] concentrated on Roman Imperial denarii, which have uniform legends, and after a geometric normalization of the text (through the use of polar coordinates) applied a HOG-like descriptor for letters. In spite of being a very interesting approach, it is limited to a very specific niche of coins, presenting difficulties in the presence of Roman numerals and being unable to deal with legends which are not arranged along the border.

The ILAC project team addressed the same problem in [55], [56] and [57], both based on the work of Wang et al. [58]. The latter allows both straight and curved words (an important feature in coins), but requires the text to have a certain size relative to the coin image size, as the SIFT descriptors are only computed at one relative size. In [59] the same team finally combines image matching and the recognition of the coin legends (using the same methods already discussed), in order to improve the robustness of image-based coin classification.

In any case, and as interesting as the ability of character recognition may seem, we can extract information from the legend only if we are given a very well-preserved coin, with very well defined legends and no other similar features complicating the "reading".

When we deal with medieval coins, we know that it is hardly an option. In fact the characters are usually so rough and worn that they are difficult to read even by the human eye. Another issue is the evolution of the alphabet styles (Latin, Uncial, Gothic, Medieval, among others) and all its regional and period variants, and sometimes the use of several of these styles in one single coin, which gives the problem a much higher dimensionality than the above approaches suggest.

3D models

The use of 3D images for recognition or identification was approached in a hybrid way by Marchand et al. [60], [61] in which several photos were taken of the same coin with different lighting directions, in order to make a model resilient to the lighting conditions.

Huber-Mörk et al. [47] discussed the advantages of using 3D data for surface analysis. Once again, they aim to avoid the interference caused by lighting variations, like shadows, or highlights due to specular reflections that distort the features of the coin. However 3D acquisitions are more laborious and expensive and, to our knowledge, 3D vision approaches applied to 3D databases of coins do not exist at the moment.

Measurement data

We have already seen that some works used sensors to retrieve more valuable data, helping in the classification process and computation time. Yet the use of sensors is not the only way to go: Herrmann et al. [62] showed a method for retrieving measurements on coins by means of a ruler placed next to the coin when taking the photo. As interesting as it is, it implies that we have control over the photo taken and thus it is not applicable in the current study.

Iconography

The project DIANA [63] avoids the usual emphasis on classification or identification, and gives us an interesting new function: it recognizes the iconography on the coin in order to map its origin. In fact the iconography on the (loosely labeled) Greek coins are very specific to the states which minted the coins, thus an analysis of its figures can give us its origins.

2.2.4 Coin Segmentation

The research on coin segmentation has seldom been approached in isolated studies, being rather associated to a whole classification process.

The already mentioned initial researches with big acclamation, like Dagobert Project [13] and Coin-O-Matic [17], set the tone to the use of a massive dataset of modern coins photographed in an extremely controlled environment. In these conditions, they need only to rely on a global threshold and some basic edge detection to provide segmentation. Many researches, as in [16], frequently working over the same dataset, made use of generalized Hough Transform (HT). The HT limitation of detecting only circles is not a problem on modern coins, so this method was broadly accepted in that context. Unfortunately ancient and medieval coins do not provide perfect circles (or no circle at all) so HT is clearly insufficient. Some pioneer ancient coins researches, as [27], proposed the use of Canny Edge to detect the border of the coin, since Sobel filters provided inaccurate edge information. Even GrabCut was deprecated in favor to Canny Edge in [52], which was shown to provide better accuracy. However, the sole use of an Edge detector (or Sobel filter) requires a homogeneous background, or its texture will present edges hard to distinguish from the edges of the coin.

In 2009 Zambanini and Kampel [31] concentrated exclusively on the segmentation problem over ancient coins and came up with a method combining Local Entropy and Local Range of Gray Values to identify the pixels of the image with the biggest amount of information, assuming that those belong to the coin. The method, still applied on grayscale images, provided good results, except when addressing the border shadows. However it assumes that the coin has more local information (entropy) than the background, which is not always the case in the present context. Our tests also revealed that Local Entropy calculation tends to be very computational intensive, even more adding a gray range calculation according to the paper description. More recently, Huber-Mörk, Zambanini, Zaharieva, and Kampel [42] presented a coin identification method heavily based on the border shape, and thus in segmentation, and suggested the use of a connected components analysis and the same Local Range of Gray Values. The method is not detailed but seems a lot similar to [31], proving the team was very confident on its accuracy. Still, in the presence of complex backgrounds, both entropy measure and local range if gray tend to highlight the background as much as the coin, making these methods unhelpful in the present context.

All these methods were using grayscale and controlled images, at least to some degree, and surely with uniform backgrounds.

2.3 RELEVANT STUDIES IN OTHER DOMAINS

Even if not specifically related to coin recognition or identification, some state of the art studies deserve to be mentioned for the potential they present to new approaches applied to coins in this or future works.

The light/shadow variation problems are addressed by Kwatra et al. [64], who present an apparently very good method to remove shadows from images. Unfortunately the process appears to be closed and patented. Guo et al. [65] presented another method, but the tests carried out in coins revealed a computation time unaffordable for this work.

Chen et al. [66] introduce the Logarithmic Total Variation (LTV) model and explain the way it removes varying illumination for face images. Although it is applied to face recognition, it could be applied in the recognition of busts in ancient coins, or major coin features.

The segmentation problem is approached in [67], where Arbeláez et al. examine the effect of multiple local cues combined into a globalization framework based on spectral clustering. Rother et al. [68] launches the concept of GrabCut, which extends the graph-cut approach by means of an iterative version of the optimisation and a robust algorithm for “border matting”.

Edge detect advances were made by Dollár et al. [69] with the use of Structured Random Forests, capable of real time frame rates (faster than most competing state of the art methods) while achieving state of the art accuracy.

Law’s Texture Energy Measure (TEM) was used in [70] on the subject of butterflies, but could as well be explored on coins.

Even if SIFT and Speeded Up Robust Features (SURF) (SIFT’s faster version) are the most well-known descriptors around, every year several more algorithms appear competing for accuracy or faster performance. A good example is given by Takacs et al. [71], presenting a Radial Gradient Transform (RGT) and a fast approximation: the approximate RGT (ARGT), which is incorporated in Rotation-Invariant Fast Feature (RIFF). They demonstrate that using the ARGT, RIFF extracts features 16× faster than SURF, while achieving a similar performance for image matching and retrieval. Other more generalized methods shall be presented in a later chapter.

3

THE PROBLEM AND ITS CHALLENGES

3.1 WHAT DEFINES A COIN

The only common element in every coin is some mark from the state or ruler legitimating its emission. That mark may take the form of a written name, a bust, or some representative symbol and usually occupies the center of the coin. Every other element may vary according to the vanity of the ruler, the propaganda message, or the stylistic taste of the artisans and epochs. Although most coins are round, there are abundant examples assuming other geometric shapes, or irregular borders. Position of the legends is not uniform, they are usually around the border, but may well be lined on the center. Figures and symbols may be representative of the culture, or commemorating some event, may send some message to the people or enemies, or can be simple decorative elements. Some variants of the same original design, as small as they may be, may constitute a different catalog reference, for instance a symbol in a different position, or a legend with a different abbreviation. All this heterogeneity is translated into innumerous variants, each one constituting a different class. As an example, Portuguese coins only from medieval period compose over 1500 different classes, most of them with just a few specimens known.

3.2 THE MOST RELEVANT PROBLEMS

By recognition we usually mean the acknowledgement of something relevant, either by realizing something as existing or previously known, or by finding some useful relation to some other known thing. So when we say coin recognition, we are entering multiple fields (or at least we should):

3.2.1 Classification

This is the most widely studied problem and probably the one that once presented to the public would be more acclaimed and scrutinized.

The act of classification consists of putting the coin successfully into a group of similar coins. These distinct groups are already established by generations of numismatists and sometimes it is not very easy to understand the criteria on which they are based. Taking into account that a group (or class) of ancient/medieval coins is not as uniform as a class of modern coins, an automatic classifier based on pictures of one's coin would be gratefully received by the whole numismatic and archeological community (as it would be lapidated for its lapses).

The challenges

The fact that they are hand minted and submitted to centuries of degradation and deformation makes it virtually impossible to find two equal coins. Their differences may be bigger due to malformation than due to class distinction. In fact the intra-class variations may sometimes stand out more than inter-class variations, it is all about conventions. To that we must add the abundant noise and deformation caused by degradation, or improper minting. So grouping coins in classes is more an approximation process than a complete match, more about the symbolism present on the coins than the visual aesthetics.

3.2.2 Identification

The uniformity and high levels of quality control on modern coins make the task of identifying a fake very hard for image recognition, unless we know beforehand which inaccuracies we are looking for. But the same factor that makes the ancient and medieval classification so hard - there are no two equal coins – is a big advantage on ancient coin identification. Identification of a specific stolen coin, for example, is a matter of simple matching. So theoretically, we are technologically fit to pick an image from eBay or some other auction house and compare it to a database of stolen coins (that is, if every museum and police force could be persuaded to cooperate). Moreover a fake coin is often made from a mold of an existing coin, so sometimes it is also possible to identify it. Yet, in this case it is not as easy as it sounds, as we will see below.

The challenges

In practice, the problem of identification is the same as classification, except that in this case each class is an individual coin, thus the volume of information and computation time will be inevitably bigger. The team from COINS project [1], [2], and later ILAC project [41] found a shortcut: instead of processing the whole coin, they concentrate only on the coin's edge. The principle is that if we could have a perfect enough segmentation from the coin's edge, we would have a faster, smaller and still accurate descriptor. The challenge is that a perfect enough outline has proven sometimes hard and it takes a long time. But the really tricky part is that the lighting conditions can severely influence the resulting extracted coin shape and the perfectly controlled lighting conditions of the dataset images can hardly be found in the real world.

Relying on the coin's edge for identification of fake coins has a very limited effectiveness since modern fakers use much more refined methods than simple copying.

Fake identification becomes a different challenge from stolen identification, in fact, three different challenges:

Falsifications by casting, the easiest and most common, are made using a mold from an existing coin, so they present a large similarity with the original coin. Yet, a mold has to have an opening to pour the metal in, in fact it has a second opening on the opposite side so the air does not get trapped. So we have already two discontinuities that can represent around 10% to 30% of the coin edge (depending on the coin size). In some cases, instead of one mold, they use two, one for each side of the coin, and then glue both halves and all the edge of the coin is filled off to smooth it, making coin edge identification even more elusive. In both cases, as the molds are used a second, a third or even more times, some imperfections start to appear shifting the fake from the original even more.

Then, there is the falsification by die-struck. In this case the faker hand-cuts a die, pretty much with the same kind of process as the ancients did, and mints a very credible coin. Even if these fakes are less common, for they require a very talented engraver, these are very dangerous fakes, often hard to spot even by experts. In this case we cannot compare the fake coin with an existing coin, so the analysis must be on the metal (like searching for silver crystallization, a sign of oldness), the style of the coin, and the search for artificial ageing processes. All these are very hard, if not impossible, to spot from a photo.

And then, there is the (so-called) fantasy coin, one that has all the correct style from the time and place it intends to imitate, but that never existed. These present day coinages mostly imitate pre-classical coins, or classes not much documented. In these cases there is absolutely no comparison point, so the identification shall be made by the same methods as the die-struck fakes, except that the stylist analysis is harder even for humans.

On top of all that, in most cases the fakers usually strive to give an ancient look, they disguise every revealing mark and force some degradation to keep it credible or to be different enough between copies so that they can sell several without suspicion. They also apply chemical patina (the characteristic colors from ancient coins) and bury the specimens to get dirty.

In conclusion, usually the fakes are different enough from the original to restrain us from using simplistic methods like direct matching or the coin's outline matching, although these can be used as a supporting process if they are efficient. The strategy should include occlusion resilience (to ignore the disguised parts of the coin) and a probabilistic approach (the more features of the coin are equal to another one, the more probable it is a fake). That said, for humans comparing a coin to every known fake is very hard to be done (as new fake types appear every day) and so the trained eye can use other signs, like too regular holes in the coin field, or small metal bubbles, or the lack of stress marks from the mint (all possible indicators of casting), or an abstract style evaluation. Even if this may seem next to impossible to do with current technology, it would be a good future course of action.

3.2.3 Style

While nearly every single study on the field seems to overlook it, computer vision applied to coins does not have to be all about classification and identification. The stylistic aspects also have a tremendous importance in numismatics research and though it is hard, with current knowledge, to evaluate a specific aesthetic facet, it is possible to compare the symbolic features between specimens. This analysis may be used for clue searching from mint origins to dating the coin relatively to others. We could go even further and try to correlate icons from geographically distant coins that could point us to unsuspected commercial or political relations.

The challenges

How do we deal with an abstract concept like style? We could evaluate the lines and motifs struck (in general, the fineness or coarseness of the struck are a good indicator of the quality of the artisan), but the edge extraction is too prone to lighting variations to make a correct analysis. The concept of aesthetics is also too hard (by now) to explain to a machine. That leaves us to symbolism. Symbols are something computer vision can deal with (such as characters, for instance), we can identify, compare and scrutinize the spatial relations between them. If we tried to build classifiers for symbols and their positioning instead of classifiers for whole coins, we would probably be surprised by the information that would give us, not only for coin classification, but also for coin class relationships.

We could also get valuable information from legend reading, but the abbreviations were so common that we would need a huge and indecisive dictionary to be able to deal with them. Besides, the medieval characters were so unpolished that it is much easier for now just to deal with them as symbols.

The big challenge is that, given the malformations of ancient and medieval coins, the current edge detectors are not very precise in these cases. We would have to assume occluded symbols and the fusion between them, and to learn which ones are real symbols and which are noise. Either we need better edge detection or better ways of learning from the current ones. This is a case that could fit well in convolutional networks, if somehow we manage to have enough images to make the method accurate.

3.3 BIGGEST CHALLENGES

Besides the specific challenges presented for each of the problems, when we look at the studies in the field, some more general challenges become evident:

3.3.1 Roughness of medieval coins

Although several researchers experienced coin recognition on classical coins (usually on Roman coins, a few on Greek coins), from all the papers searched only two teams approaching the problem with medieval coins could be found, and both in a sideway, comparing with modern or Roman coins and achieving modest results. The difference is not as subtle as one may imagine. In fact, after the fall of the Roman Empire, the artistic quality standards and the artisans' talent gave a massive leap back. The coins from medieval times were as coarse as the ones thirteen centuries before (this is particularly true for peripheral European states). So the task of recognizing medieval coins is expected to be even more difficult than the already approached classical coins.

It gets worse: the chosen datasets, almost always the same ones, come from museums' collections and represent coins generally with a very good grade (conservation status). If we try to deal with the coins generally found in the market, private collections, archeology sites, or minor museums, we will confirm that such a grade of coins represent only a very small fraction of the specimens available. Most coins present a modest conservation status, which means an even harder task on relevant features extraction.

3.3.2 Uncontrolled coin images

While on laboratory one can build a dataset in a controlled environment with uniform conditions. When out in the general public, that's an impossible premise to maintain. The common images shared on the internet present a dreadful diversity of: light colors, directions and intensities; shadows, highlights and reflections; complex backgrounds; sometimes the picture is taken with the coin inside its protective blister; the image resolution is at times too low, the noise too high; and a lot of other variations we cannot control. So, in any problem we shall focus on, the first step must categorically be trying to find the most resilient algorithms and homogenize as far as we can all these variations.

3.3.3 Small data

When we read in a scientific magazine, or on the internet, on the subject of computer recognition, one concept is omnipresent: Deep Learning. Big Data, Convolutional networks, and alike notions seem to be putting a test on every imaginable problem around. Yet, we cannot speak of big data when so many coin classes have no more than a handful of existent specimens, and often only one or two photos around. One may argue that if they are so rare the problem of classifying is not so hard, but the variations between classes are often so subtle that it is problematic to distinguish if we have two coins of different classes or if it is some variation inside the same class. If we want to classify a coin, we must have classifiers; however we can hardly have a complete dataset, and even that will be scarcely populated for each class. Therefore, instead of talking about Big Data, in this context we should be discussing Small Data.

That said, it does not mean we should not use the internet communities to help feed the learning machine, on the contrary, their help is crucial in obtaining a big database and correct its mismatches, towards a recognition system both robust and helpful. That is why it is so important to bring these tools to the public instead of keeping them in the laboratory.

3.4 PROPOSED APPROACH – SOLUTION

3.4.1 The goal

The aim of this work is to contribute to a model for automatic recognition of medieval and ancient coins that could lead in the future to a system opened to the big public and helpful in classification and identification problems.

It is not the purpose to create new algorithms or methods, but to use or adapt the current state of the art, for only that way it is possible to cover such a large scope. That said, some innovations were needed in order to cope with the rough initial premises.

Since the goal is to orient the model to big public usage, there are two premises:

- The software, methodologies and algorithms are preferred to be of free access, or open source. This choice implies the dismissal of methods which hold a patent, closed, or with no available implementation, as promising as they may seem. It does not mean proprietary software cannot be used during the tests, but the final model must be free to use or be able to be adapted for that purpose.

- It must be oriented to the use of non-controlled images, as it is the expected reality. Nevertheless it was needed to include controlled images during the tests in order to understand the most critical variances and how to deal with them.

The work includes an especially collected image dataset of medieval Portuguese coins. This intends to contribute to a publicly available dataset for future research, and hopefully inspire other colleagues to focus on this area. The choice of this dataset does not affect the outcome of the tests, since the Portuguese medieval coins, especially the ones from the first dynasties, are among the coarsest in Europe, and so the starting point is kept as hard as possible as it is the objective. Nevertheless the tests are also done with a known dataset of Roman coins to provide a comparison point.

3.4.2 The approach

The research is divided into two parts:

- Segmentation, addressing the background removal and optimization of the uncontrolled images. The purposed segmentation method is divided into four stages:
    - Stage A.1 – Histogram Backprojection
    - Stage A.2 – Border approximation by Convex Hull
    - Stage B.1 – Canny edge
    - Stage B.2 – Refined border

- Feature Extraction & Recognition, addressing the mathematical description of the features of the coin and its classification/identification, it is divided into:
    - Feature Description
        - Feature Extraction
        - Feature organization into Bags of Visual Words

- Classification
  - Machine Learning Training
  - Machine Learning Testing

Segmentation of non-controlled images

As the chosen subject and conditions are very hard, this work concentrates more on this initial step of the recognition process since it presents the biggest challenge and the most ill-studied one.

Dealing with non-controlled images implies minimizing the effects of poor lighting choices, with cast shadows and highlights, as well as poor resolution or highly compressed images which result in pixilation, as all these factors interfere in a very negative way with the correct edge detection and feature definition. An even harder obstacle are the complex backgrounds often chosen, which may be very difficult to separate from the coin due to the background's texture, variable colors or multitude of different areas. As an example, it is common to see images with the coin on the hand, or over a fabric tray, or inside a protection case. To correctly classify a coin one must be able to ignore these backgrounds, or their features may be taken as coin features, thus providing wrong results.

Figure 1 - Examples of hard images to segment

(Left to right: a case of too high compression and pixilation; coin inside a holder with bright reflections and distinct background areas; a coin on the hand and with severe cast shadows)

After testing numerous state of the art techniques, Histogram Backprojection was chosen as the base algorithm to separate the coin from its background. Several optimization techniques and filters detailed in chapter 4 are used to minimize the stated negative variables, and Canny edge filter helps refining the segmentation.

Feature Extraction & Recognition

The knowledge from the optimizations tested is also useful to improve feature description in the second part of the method: the recognition. For this task, mathematical feature descriptors are extracted (like SIFT, SURF or DAISY) and subsequently they are organized into Bags of Visual Words in order to provide higher level, more meaningful descriptors.

Machine learning is used to train a set of labeled images and, given their high level descriptors, establish a model capable of separating them into the respective classes. This model allows a new unknown image to be rapidly tested against the classes in the search for a match.

Several different feature descriptors (SIFT, SURF, DAISY) and machine learning methods (SVM, Random Forests, Naïve Bayes, k-NN) are tested and compared in the search for the most accurate method. The process is detailed in chapter 5.

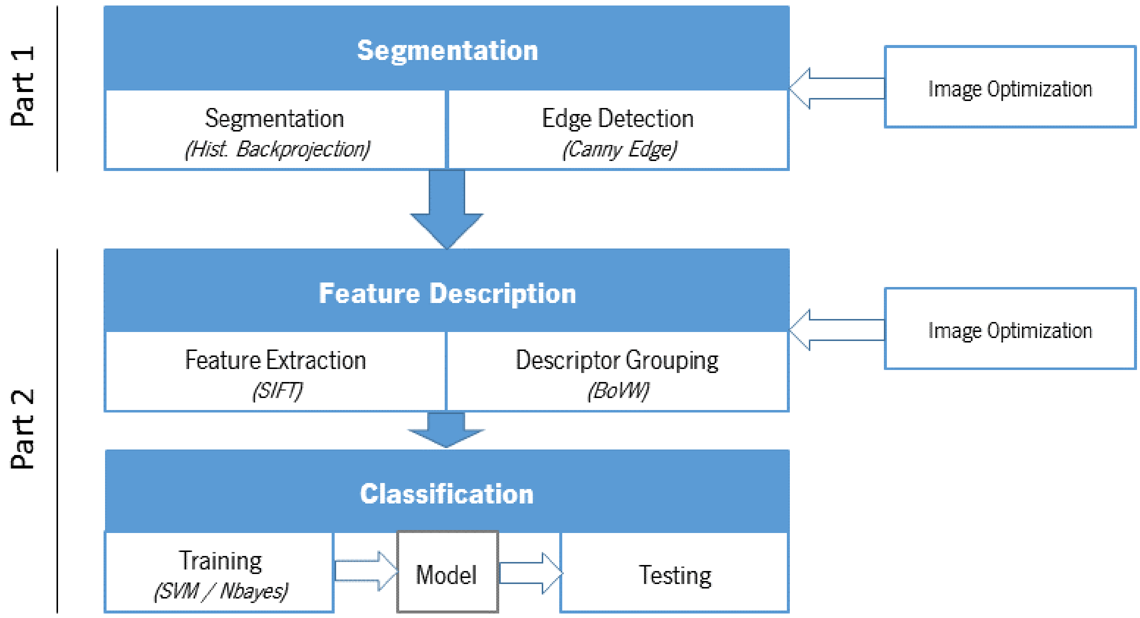

Figure 2 - Overview of the proposed approach

As discussed before, both identification and classification should be made by a probabilistic approach, being the big difference a matter of building class descriptors or specimen descriptors. In that sense both methodologies can be approached in the same way.

As for the style analysis (in the previously explained sense), this is a virtually unstudied concept, with its own challenges demanding new approaches, so it deserves its own focused research and it is not to be covered here.

Neither character recognition nor facial detection were used to assist the recognition process because, as discussed above, their contribution would be too limited in these particular coins and uncontrolled images.

3.4.3 The framework and test environment

Even if Matlab is the most widely used framework in the research community, this work was based on OpenCV 3.1 over C++. This is an open source framework, that besides going along with the initial premises of free access for all, it makes much easier if someone wishes to import the models here defined and programmed to end user application (either web, desktop or even smart-phone based). As a bonus, some studies suggest that OpenCV tends to be faster than Matlab dealing with image processing, as shown in [72].

As for the tests, a controlled environment image dataset of medieval coins, detailed in chapter 4, was composed in order to represent the most common variations that interfere with the segmentation or edge detection, as: coin color and texture; lighting color, direction and intensity; noise and poor resolution; shadows; and different complex backgrounds. To this set some challenging images chosen from the internet were added. This dataset was the basis for the tests over Segmentation and Edge Detection.

For classification/identification using machine learning algorithms, a known existent image dataset of Roman coins [73] was used, in order to have a comparison point. At the same time, as mentioned before, a dataset of Portuguese medieval coins was assembled to represent the same tests on medieval coins.

All the tests were performed in a laptop with operating system Windows 10, 64 bits; CPU Intel Core i7-4720HQ 2.6Ghz; 16GB of RAM memory; and a graphics card Intel HD Graphics 4600.

4

METHOD: I – SEGMENTATION

4.1 INTRODUCTION

Segmentation, when applied in the context of coins, usually refers to the separation of the coin from the background on an image and it is expectedly the first step in any classification method, otherwise the computer may take features from the background as belonging to the coin, thus interfering with the correct prediction of its class. In a simplistic example, several coins photographed over a same textured background could be wrongly taken as belonging to the same class.

Removing a complex background may be a very hard and unpredictable task. That is why the researchers keep basing their work on little more than a couple of available image datasets, with grayscale pictures taken in carefully controlled light conditions, and contrasted backgrounds in uniform tones, in order to maximize detection results. This presents a serious problem in two ways. Firstly, we are wasting vast repositories of coin images all over the internet that could be used to make our algorithms more robust, and secondly, the same algorithms that achieve great results on these controlled images, when applied to “real world” coin images, tend to be very disappointing.

It is undeniable that dealing with internet or uncontrolled images and their low quality standards brings great challenges. Dark cast shadows or bright reflections severely interfere with most algorithms. Poor image resolution or too high compression lead to pixilation or “block” effect, thus adulterating the shapes. Ancient and medieval coins provide even more difficulties, like the irregularity of its borders that very rarely are a perfect circle; some coins are so thin that they barely contrast with the background; some are so thick that their shadows extend their border on segmentation algorithms; some coins are square, some are cut in half, some have holes. To make it even worse many images display the coin inside protection holders, making it very hard for a segmentation algorithm to distinguish between the coin border and the holder border. On the upside, the computation power is now much higher than a decade ago, so there is no more reason not to use color information on our algorithms.

Although the homogeneity of the controlled datasets facilitates higher (apparent) success rates, this is a limitative premise. The segmentation of the public coin images, overcoming big problems as bad illumination and terrible background choices, is the ground to more robust approaches on classification, not only narrowing the gap between researchers and public applications, but also allowing easy access to huge image datasets.

As this is the most ill-studied step on the present context, it is also where this research may give the most significant contribution, and so it is where the biggest share of time and effort were thrown in order to achieve a new efficient method.

A primary approach of the present method was accepted in the 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC 2016) and gave way to a paper to be published by IEEE [3]. Meanwhile the research continued and the method has evolved to include edge information on the detection, as long as some optimizations.

4.2 PROPOSED METHOD

Despite its reliefs, a coin usually presents a narrow range of colors. Even if on occasion ancient and medieval coins may present areas of oxidation or dust, usually the most dramatic color variations are on its intensity. Thus, the proposed approach begins with isolating the coin from the background based on a probabilistic comparison of color hue and saturation using a Histogram Backprojection algorithm (stage A1, described in chapter 4.2.1). Ignoring the intensity value has the additional advantage of diminishing the shadows noise on the border detection. We can then calculate a border approximation by means of a Convex Hull over the previous results (stage A2, chapter 4.2.2).

Some optimizations, detailed in the following chapters, are made along the process in order to reduce noise and enhance the borders. The edge operators are particularly useful in the presence of pixelation caused by less than optimal compression of the original image.

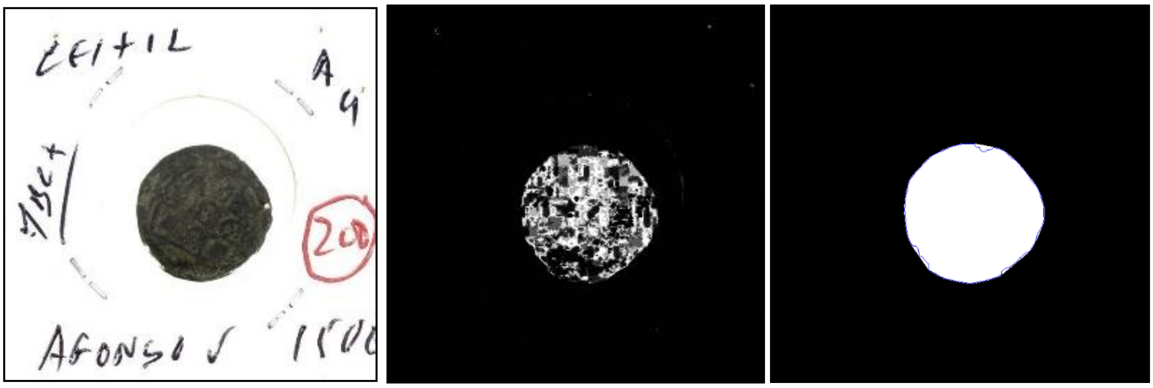

Figure 3 - First stage of Segmentation

(Left to right: original image; HB result; calculated mask)

A following stage uses edge information in order to refine the detected border. This stage is beneficial especially in cases where the texture of the image is similar to the texture of the background. A Canny edge operator is calculated over the original image, and limited to the ROI by application of the previously calculated mask (stage B1, chapter 4.2.3), a subsequent Convex Hull (stage B2, chapter 4.2.4) results in a new mask to apply over the original image hopefully removing all the background.

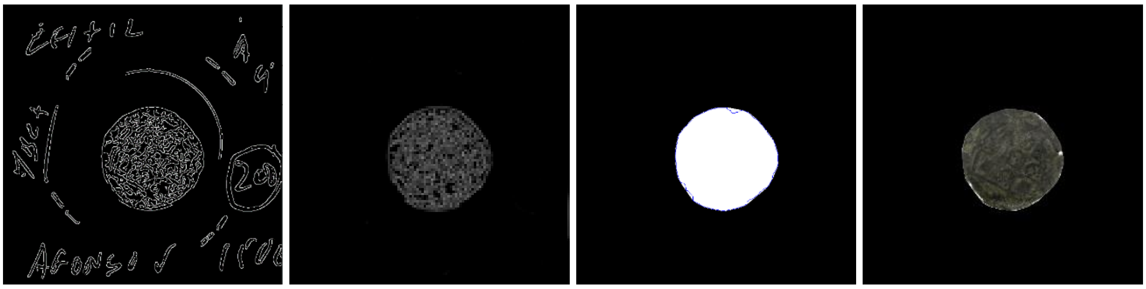

Figure 4 - Refining segmentation with edge information

(Left to right: Canny Edge of original image; First Mask over Canny; final mask; mask over original)

The complete process, with all its stages and examples of its outputs, is outlined in Figure 5. The following sub-chapters describe in detail each of the stages.

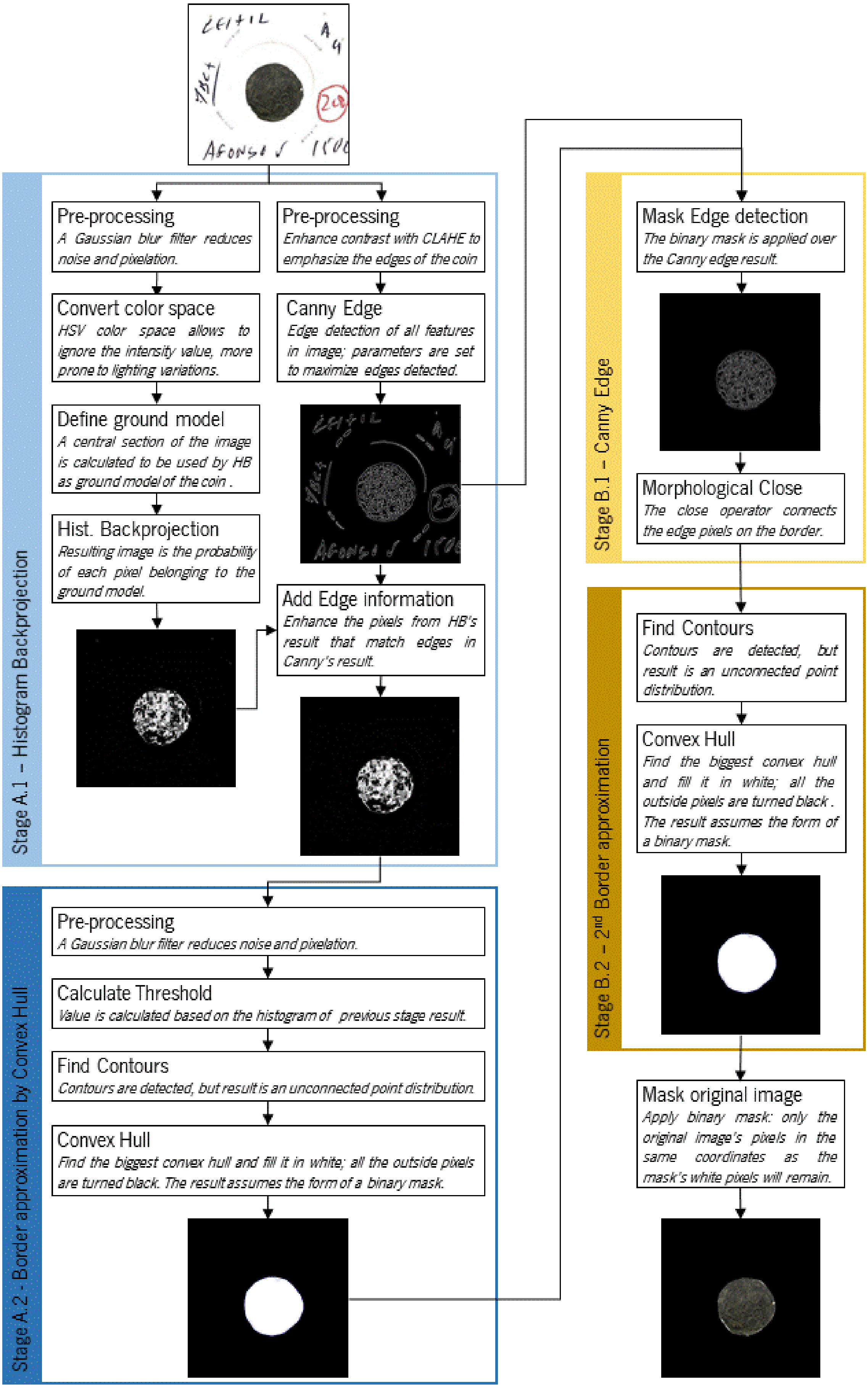

Figure 5 - Proposed method for segmentation.

4.2.1 Stage A.1 – Histogram Backprojection

Histogram Backprojection (HB) has been proposed by Swain and Ballard [74] back in 1992. Given a model image, the HB algorithm computes its histogram and then back-projects it in the test image's histogram in search for fitting pixels. The result is a single channel matrix with the probability of each pixel belonging to our ground model. In other words it gives us the probabilistic prediction of our object of interest.

Commonly HB is used with a previous model image like a section of skin to detect faces or hands in a test image. The test image may also be used as the model to perform some kind of self-HB, assuming that the object of interest fills most of the image. Given the variability of the coins, uncontrolled images and the complexity of possible backgrounds, neither of these approaches by itself is adequate in the present context.

The stage here called Histogram Backprojection includes some pre-processing, an automatic model definition, the HB algorithm, and a refinement using edge information.

Pre-processing and Color space conversion

A previous filtering with a Gaussian blur has shown to improve the results by removing noise. Gauss blur proved to be more successful than edge-preserving filters like Median blur or Meanshift.

Although a following addition to the image of its Laplacian, thus sharpening the image edges, may sometimes ameliorate outcomes, this was deprecated in favor of an edge refinement over the HB discussed later.

Converting the image from the usual RGB color space (Red, Green, Blue) to HSV color space (Hue, Saturation, Value or intensity) allows us to ignore the intensity value, more prone to lighting variations, and rely solely on the hue and saturations to define our object of interest. Obviously, a grayscale image, which has only one channel based on the intensity – the channel we desire to ignore -, provides us low discriminating capabilities and it is not a good candidate for this algorithm.

Defining a ground model for Backprojection

The automation of the process implies being able to calculate a good ground model for the HB algorithm, this will be the template to which every pixel of the original image is compared in order to determine if they belong to the coin or not.

The tests revealed that using a partial section of the coin in the test image results in a good model for the segmentation. The larger the section model chosen, the better it covers the tone and shadow variances of the coin, thus tending to provide a more exact result. Nevertheless we cannot risk to overflow too much over the background or to choose the wrong section. In order to automatize the process, a rigid rectangle model with the following characteristics is established:

- centered in the image;
- sized on 20% of the image dimensions.

These conditions may be changed according to the needs of each project, yet they are not arbitrary: Even if most coins are round, using a round model would cover only its center area, thus leaving out most nuances caused by lateral lighting (shadows and highlights). The rectangular shape allows us to cover the center and to reach out a part of the outer sections of the coin with the model's corners. Also the position and dimensions of the established model force that the coin is also centered in the image and occupies at least 20% of it. Most images will easily satisfy this requirement, yet, in the cases where the coin is smaller, or there are several coins, or (the most common case) both sides of the coin are collated on the same image, we can use Cascade Classifiers, or any other detection method, to identify each coin and treat them separately, one at a time (this detection step is not covered in this research).

Figure 6 - Original image with rectangle model (left), result from HB (right).

Histogram Backprojection algorithm

Having a ground model as seed, its bi-dimensional histogram (for channels Hue and Saturation) is calculated. Then HB algorithm computes the probability of each element value in the original image in respect with the probability distribution represented by the histogram. In other words, the resulting matrix has the same size as the original image and each pixel values shows the probability of the correspondent pixel in the image to match the ground model.

This resulting matrix may be shown as an image, as in Figure 6, and has proven very efficient in attenuating complex backgrounds. However, sometimes the border contours may seem roughly defined, thus the need to refine the result.

Adding Edge Info

Hue channel is particularly sensitive to artifacts caused by less than optimal image compression (in jpeg for instance). This may lead to an extravagant aliasing on the HB result. In these circumstances the border may be refined using edge information from a Canny edge [75] operator over a grayscale version of the original image. The Canny edge operator is further addressed in stage B1 (chapter 4.2.3), where it will be reused.

We cannot simply add both results (HB's and Canny's) because it would retrieve back all the undesired features or noise. A *bitwise And* between the results is a better option, but still less successful than the chosen one: inflate the HB's pixel values that coincide to the edge detection by a multiplication factor. For instance: if some pixel in the HB result has the value 15 (dark, meaning low probability of belonging to the object of interest) and this pixel is in the edge detected by Canny operator, then we multiply that value by, let us say 4 (tests showed using values 2 to 4 as multipliers to have the best results; with bigger multipliers the noise becomes too significant), getting a new value of 60. Our pixel will now for sure be kept once we do the binarization by thresholding on stage A2 (chapter 4.2.2).

This process does not add border information where it absolutely was not found by HB, on the other hand it will also not highlight noise or undesired features to a degree that will interfere with the thresholding. It is mainly useful when the texture of the coin is similar to the texture of the background. In the remaining cases the tests showed that even if the improvements are very slight, they are relevant enough to include this stage in the method. The increased processing time is not significant, since we will reuse the Canny edge calculation on stage B1 (chapter 4.2.3).

4.2.2 Stage A.2 – Border approximation by Convex Hull

Pre-processing

A Gaussian blur is applied to the output from the previous stage in order to weaken noise peaks.

Automatic Threshold calculation over HB

A threshold is decisive to eliminate the residual noise that still populates the background of the image, and to better define the shape of the coin's border.

A good estimation of the most favorable threshold value possible over the HB results is a crucial point in the presented method, but traditional automatic methods like Otsu's or Adaptive threshold failed miserably in this task.

On empirical analysis, a relation was found between the ideal value and the end of the gradient from the biggest histogram peak: we reach an ideal value being equal to the first histogram bin, after the biggest peak, that is lower than 2.5% the maximum histogram value (i.e. if the maximum histogram value is 400, the ideal threshold value shall be the following bin number with a value lower than 10). To prevent remaining noise, the next few pixels are also required to remain below that percentage.

Even if this value fails in some cases, it was the best approximation found, beating all statistical calculations (like averaging and standard deviations) and the mentioned traditional methods. In Figure 7 we can see an example of HB's histogram (flatted for display purposes): the maximum peak usually occurs in the first bins, representing the low probability assigned to background pixels. The secondary big peaks shown in this example suggest a somewhat troubled result with remaining noise.

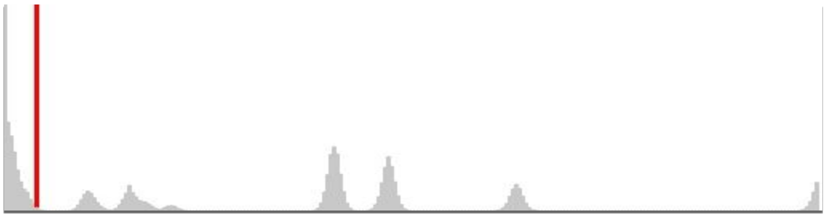

Figure 7 - Histogram from HB result with a red line on the estimated ideal threshold value.

After the threshold the background should have been almost completely replaced by a black level.

Find contours and Convex Hull

By then whatever remains must belong to the coin. An algorithm to find contours is applied, although the result is presented as an unconnected point distribution and not a uniform shape. To find the outline we calculate the Convex Hull, which is defined as the smallest convex set that contains a set of neighbor points. If more than one shape is found, either inside the coin (from shadows or colored dirt on the coin), or outside (from remnants from the background), the biggest one existent is chosen and taken as the outline (border) of the coin.

As a failure recognition system, if the biggest shape's area is smaller than 20% of the image area (the size of the model rectangle) then it is marked as an assured failure and the chosen shape is set back to whole-image, based on the consideration that it is better to have a non-segmented image than just a small fragment of the coin. This possibility is interesting since it allows the automatic process to try segmentation again with the Canny Edge stage over the initial image, or to use a completely different method, without the need for human interaction.

The outline calculated (the biggest shape) is filled in white, and all the outside pixels are turned black. The result assumes the form of a binary mask. This mask when applied over other image will retain only the image's pixels in the same coordinates than the mask's white pixels. In other words: the background will be turned black.

4.2.3 Stage B.1 – Canny edge

If we really need real-time performances, we can end the method here (and even deactivate the previous *Adding Edge Info* stage). But we can significantly improve accuracy by refining a bit more the segmentation, especially in the cases where the first stage failed or had less than optimal results. As we based the first stage on color/texture information, it makes sense to base this stage on edge information.

Canny Edge

The algorithm for this operator was proposed by John Canny [75], and intends to find the edges of an image by a multi-stage algorithm consisting in: applying a Gaussian filter to remove noise; finding the intensity gradients; applying non-maximum suppression; applying a double threshold to determine the edges; and tracking the edges by hysteresis for suppression of the edges that are weak and not connected to strong edges. This is still one of the most robust and widely used algorithms for edge detection.

As a way of predicting the best threshold values for Canny edge, Fang, Yue and Yu [76] suggested the use of the Otsu's threshold value as Canny's high threshold (T1) and half that value for the low threshold (T2). This choice proved particularly efficient in images whose histogram presents the characteristic of two-extremum, which is rarely the case in pictures from coins. Nevertheless, the tests carried out in this research's context corroborated that Otsu's value is a good reference, yet more detailed results came from using *T1 = Otsu/2* and a *T2 = Otsu/4*. This values were later corroborated by [77] that presents a very proximate conclusion (for T2 they use *0.3*Otsu*).

In another line of thought, we could also use the average (avg) of the image values and its standard deviation (std), in the form of: *T1 = avg/2*, and *T2 = (avg/2) - std*. Both estimation methods result in very similar values, and show no perceptive differences.

Masking the calculated Canny edges

As good as Canny's algorithm may be calculating the edges, it provides no means of distinguishing between our coin and the unintended features or background patterns. Thus we apply the mask calculated in the first stage.

Morphological operator: Close

After we apply the Canny edge operator, the result will be an image with black background and white features' edges. Still, rather than lines delineating the edges, we have a group of unconnected pixels. We can apply a common *Find edges* procedure to connect these pixels, but there is a problem: on uncontrolled images, we want to choose Canny's threshold values that maximize the edges in order not to end up with an open contour of the coin, but this also means that the inner edges from the coin faces will also be maximized. As these inner edges often touch the border of the coin, the *Find edges* procedure often follows them causing a contour with canals cutting the outer border into the inner coin.

Applying a Close operator before finding the edges minimizes to a great extension the mentioned problem. On the downside: sometimes some small holes or dents persist on the border, inhibiting us from using the direct edge found.

4.2.4 Stage B.2 – Refined border

A second Convex Hull

Next stage recalculates a convex hull over the newly found edges. The process is the same as before, except this time we do not need to apply a threshold because the image is already binary. Again the biggest convex hull is chosen and it is filled in white while the remains are turned black.

4.2.5 Final result

The result is a mask that allows the removal of the background of the original image replacing it by the color black. It is important to notice that all the filters and processing made to the image during segmentation are temporary and exclusively to achieve the final mask. This mask is applied over the original image in its original state, thus the resulting image is the exact copy of the coin portion of the original image over a black background.

Figure 8 - Original and Segmented images

4.3 EXPERIMENTS AND RESULTS

As the segmentation over non-ideal images is the least studied step of the classification of coins, this research invested a vast amount of time and effort being thorough on studying and testing the existing methods, either basic or state of the art, always respecting the initial premises. The proposed method evolved over the limitations found along those tests. In the hope that it may be useful for other researchers to avoid the same errors, a list of other main methods and variants tested is presented in the Support Material chapter, along with some simple notes over their advantages and disadvantages.

4.3.1 The datasets

Development dataset

In order to identify the variants that negatively interfere with the coin segmentation, a dataset was constructed over 17 medieval coins, chosen to represent the combinations of their usual relevant characteristics: color, brightness and field texture. Over these coins we constructed a controlled data set contemplating hard conditions, with combinations of the following variations: lighting directions, shadow intensities, shadows over fractures (over several directions), textured backgrounds (colored coin trays, wood, marble, textured metal and newspaper) and noise (either caused by bad exposure, low resolution, or bad white balance). A small collection of 18 images of unusual specimens, like oddly shaped coins (square, oblong, holed, or concave), half coins, or unusually bright ones, was combined to the dataset. Over the refining tests the images that did not represent real variations, or repeated similar cases, were removed. In order to cover non-predicted variants, a random set of 32 coins was selected from the internet, regarding their complex backgrounds or challenges.

The resulting dataset, although partially having controlled conditions (in a harmful way), represents a collection of harsh cases that were used to study the challenges and to refine the proposed method, hoping to build a system as robust as possible.

Test dataset

To avoid the common pitfall of over-tuning a method to a dataset, a second collection was built of random internet images that provided a final test dataset.

A spider was used to transfer images from a google search on “medieval OR ancient coin”. After removing all unsuitable images (like images with several coins, drawings, or unrelated images) and the ones that did not follow the premises for both methods (the proposed method and GrabCut), the first 100 images were chosen to compose a test dataset. This provided a random collection of “real world” images, used for a final testing.

4.3.2 Comparing method - GrabCut

As a comparison method we used the general purpose GrabCut algorithm [68], an approach based on optimization by graph-cut. This interactive method relies on a user defined bound box around the object of interest. Then it estimates the color and contrast distributions from both the object and the background, isolating the connected regions inside the bound box. The process may suffer multiple iterations until the user finds a satisfactory result. Some interesting variations of this algorithm appeared meanwhile, including “GrabCut in One Cut” [78], which is interactive just the same but, as the name indicates, tries to achieve the best result in only one iteration.

In order to automate the process an arbitrary bound box was chosen, 10 pixels from the image edge, on the assumption that people always leave a margin around the coin. If one wants to limit the method to round coins, one could also use the corners of the image.

4.3.3 Results

The results of both methods applied over the Test Dataset (100 random google images) are summed up in Table 1 and are grouped according to four criteria:

- Good: when the segmentation follows closely the border of the coin.
- Acceptable: near miss, leaves a small margin or small bits of the background around the coin, but still provides an acceptable segmentation.
- Non-acceptable: near miss, similar in magnitude to Acceptable, but crops minor segments of the coin possibly limiting a future classification process. In other contexts both Acceptable and Non-acceptable may be joined into one acceptable/failed group.
- Failure: when the detected outline crops significant segments of the coin, or leaves large areas of the foreground.

A qualitative evaluation inevitably falls on some degree of subjectivity, yet this scale was preferred to an exact quantification of the deviation for the reason that it would not improve the usefulness of the evaluation. In fact, if we have a margin of a couple of pixels along all the coin border, the deviation would be the same as an equivalent area cropped from the coin. In the first case the segmentation would still provide a very valid object for classification, in the latter it would seriously damage the attempt of classification.

Figure 9 - Samples of the classification criteria

(Segmentation represented as a red line in order to facilitate the evaluation; Left to right, top-down: Good; Acceptable, Non-acceptable; Failure)

Method	Segmentation results			
	Good	***Near miss***		***Failure***
		acceptable	***non-acceptable***	
GrabCut	30%	19%	07%	44%
Proposed	51%	25%	09%	15%

Table 1 - Summary of empirical evaluation results

The average computation times are presented in Table 2. For a better perspective, the images were also separated into Small (file size bellow 400KB, average size of these images: 83KB), Medium (between 400KB and 1MB, average size: 580KB) and Big (above 1MB, average size: 2,1Mb) and time averages were calculated for each of the groups.

Method	Segmentation times			
	Average	***small images*** *average: ~80KB*	***medium images*** *average: ~580KB*	***big images*** *average: ~2,1MB*
GrabCut	19.58 s	9.63 s	82.66 s	148.65 s
Proposed	0.29 s	0.16 s	0.80 s	2.62 s

Table 2 - Computation times on empirical evaluation

4.4 DISCUSSION

None of the methods provide next to perfect accuracy in border description, so their use for cosmetic purposes is limited. That said, the goal of this work is to provide a suitable coin segmentation to be followed by a classification or identification process, and in that perspective the outcomes are very satisfactory.

The chosen GrabCut approach presents reasonable results, and it easily addresses coins' fractures and holes. Surprisingly, while in the development dataset (the "hard cases"), GrabCut results fall mostly in the class of Near misses, in the test dataset the results tend to be more decisive and fall in Good or Fail. One limitation is that, although it works well with complex backgrounds, such a large initial bound box may embrace undesired elements that will not be excluded from the result image. In general terms, this method is very sensitive to shadows and tends to include them in its segmentation misstating the results. But the biggest deal breaker is the elevated computational times, typically from 10 seconds (for small images) to 150 seconds (for larger images), which may stand prohibitive in many end user applications.

The proposed method achieves a good segmentation in half of the internet images and a low failure rate of 15%. Its biggest challenge is when the coin tones are very similar to the background. It also does not address the concave curves on the coin border, including fractures, due to the use of the convex hull method.

Both GrabCut and proposed method results may be affected by extreme cases of compression artifacts, the blocky effect caused by some algorithms like jpeg when too high compression is applied. This is something hard to attenuate and no process was found to overcome this problem.

The proposed method reveals to be very efficient dealing with shadows and highlights, and excluding them from the segmentation result. Besides accuracy, a very significant advantage of the proposed method is the computation time, in average more than fifty times faster than GrabCut.

5

METHOD: II – FEATURE EXTRACTION AND RECOGNITION

5.1 INTRODUCTION

The previous chapter helped purge our tests images from everything but the coin itself, but since the beginnings of machine vision we know it is not possible to match two distinct images (even from the same object) by a simple bitwise comparison. There are too many variants – like lighting direction or color, or intra-class variations– for that to be a valid solution.

As humans, in order to classify a coin, we find its meaningful elements (or features) and compare it to previously known elements from other coins. The classification process is a matter of finding the class with all the same elements (or at least most of them in case there is occlusion or poor conservation). Dealing with ancient coins, we tend not to be too picky about the shape similarity. In an over-simplified example of a common symbol on coins, a cross is a cross, no matter if its branches are a little thicker, or if it is damaged in a corner.

Machine recognition shall follow the same principles: we need a higher level description of our object. More than that, we need a way to deal with all the variances that description may have, either environmental inconsistencies (again: lighting, angle, among others), or intra-class variations. Ideally the description should also be orientation-invariant and scale-invariant, since the non-controlled images do not guaranty a uniform scale or a correct orientation, and also being able to deal with occlusions, making it tolerant not only to real occlusions, but also to too aggressive segmentation crops or simple worn out of the coin.

Having these feature descriptors, we then need to divide groups of features into classes. This task may be fuzzier than it seems. In the above example, we must admit that there are many different types of crosses (Latin cross, Christ cross, Templar cross, and tens of others) and sometimes it is not very easy (even for humans) to discriminate between them, especially if they are not in perfect shape. Although initial researches used a simple shape compare (suitable for modern coins), that proved too rigid for ancient coins. We need some advanced techniques of machine learning in order to be able to separate sets of features (with many shared elements) into different classes, and then be able to apply that skill into new images to classify them.

The difference between a classification problem and an identification problem is in the magnitude of the classes: while in the former, each class groups several specimens, in the latter each specimen is a class. A direct implication of this remark is that the identification problem is in fact a simplification of a classification problem. Consequently, in this research we will focus on the classification problem and assume that identification can be achieved just as successfully with the same method, or, as an alternative, may be derived by a simplification of that method.

5.2 PROPOSED METHOD

In the first part it was shown how to remove distracting backgrounds from the coin's image. This is important in the classification process because a machine could easily confuse patterned backgrounds with features from the coins, resulting in grouping in the same class several coins photographed over a same table for instance. Therefore for the classification process we must assure an image where only the coin's features are relevant, either by photographing it in a controlled environment over a uniform background, or by segmenting the image.

The following sections provide an overview over the stages and concepts applied in the proposed classification method followed by detailed explanations on each stage.

5.2.1 Overview

From the overview presented on Figure 10 (with the processes in blue boxes and data results in white) one can immediately see that the classification method is divided into two main stages:

- Training stage: when the machine learns how to distinguish between classes.
- Test stage: when the machine predicts a class for a new and unknown image.

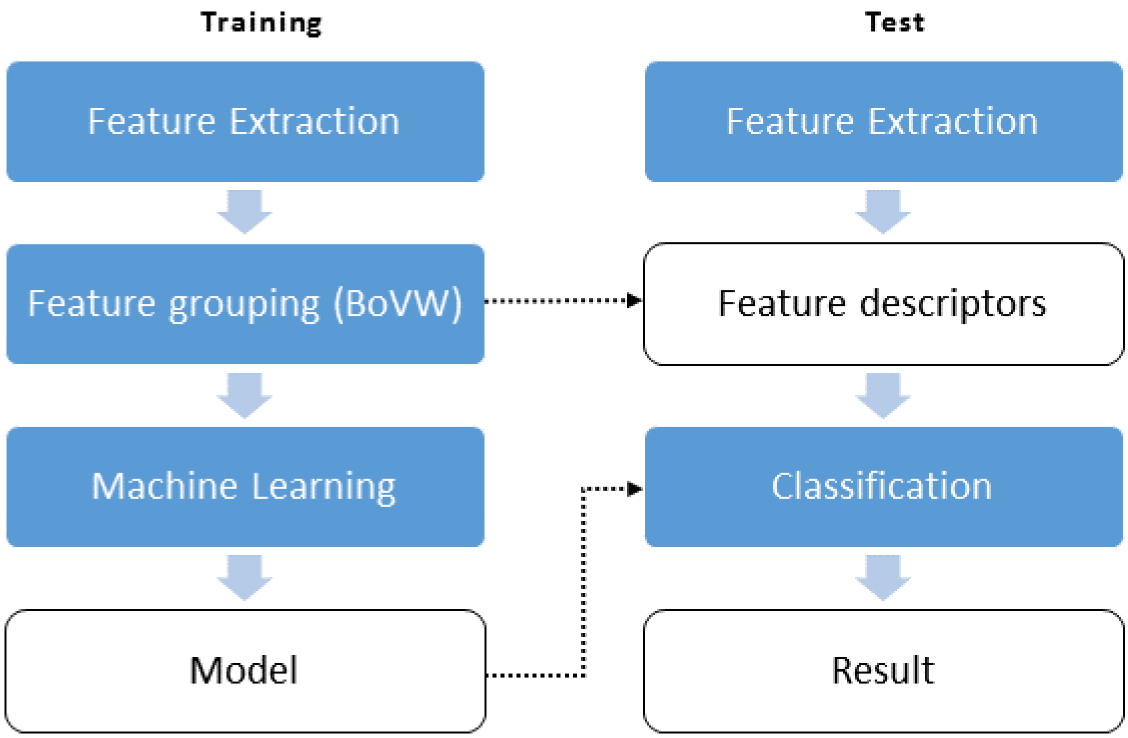

Figure 10 - Overview of the proposed classification method

Feature extraction using descriptors

The proposed method starts by extracting the features that allow to describe an image. As explained above, bits are not enough to define features, so we need a higher level description of our object, and one that follows at least these two conditions:

- Focus only on the meaningful features from the object;
- It is able to translate that description into a mathematical language, so a machine can deal with it.

Many visual feature descriptors in compliance with the above conditions have been proposed in the past, from simple shape or color to more elaborated ones like texture, energy or more abstract forms. In this research some of the latest and more successful feature descriptors are tested, SIFT and SURF, as long as an open source non-binary descriptor: DAISY. These are further detailed in the following section.

The extraction of the descriptors from an image consists in calculating the chosen mathematical descriptor for a given set of key points (key pixels) on the image. Often the implementations of feature descriptors, as is the case for SIFT and SURF, include a

Detector to identify the most promising key points. Theoretically this allows to avoid wasting computation power in less significant points. Using all the pixels, or a subset of them (for instance 1 pixel on each 10), as key points increases the computation time but in some cases provides a more distinctive description of the image. This is generally called a Dense approach. Both procedures were included in the tests.

Bags of Visual Words

Just as letters have a much stronger meaning when grouped into words, the descriptors extracted will then be grouped into Bags of Visual Words (BoVW). This grouping intends to cluster distinctive sets of features in order to increase descriptors significance and provide a more effective description of a given coin. After this stage, the features, although retaining the designation, are in true groups of features.

Machine Learning

After achieving a robust description for each image, we need a way to teach the machine how to distinguish to which class it belongs to. That task is well suited for a supervised learning approach, one that conjectures a function from labeling training data (in opposition to unsupervised learning where data is unlabeled, thus excluding the possibility of evaluating the solution). For that we need a set of labeled template images, the labeling being the class which each image represents. That set is then “trained” by a chosen supervised training algorithm in order to provide us with the function (or model) that will allow us to classify future test images. The chosen algorithms to test and compare in this work were: Support Vector Machine (SVM), Random Forests, Naïve Bayes and K-Nearest neighbors. After this training, one can test a given image against the model in order to predict its class.

The following sections shall explain and justify the choice of the methods that were included in the tests in each of the stages presented here.

5.2.2 Feature Extraction

The previous work on this area, summed up in the chapter State of the art, shows that the latest and most successful articles tend to use SIFT, or its faster variant SURF, for detection and description in feature extraction, thus these methods must be included in this research. In the last few years many alternatives for SIFT were presented, however we need to restrict to feature descriptors that, just as SIFT and SURF, are both:

- Scale-invariant, because the coin images are taken in no controlled environments they can be presented in different scales; and
- Rotation-invariant, because we need to classify the coin in any position it is presented and we cannot assume a "correct" position.

Unfortunately there is one more restriction due to limitations on the platform used in this research: many implementations of BoVW, including OpenCV v.3.1, cluster the key points based in Euclidean distance, which tends to originate wrong clusters when dealing with binary descriptors like ORB [79], A-KAZE [80] and some other promising feature descriptors. An work-around could be implementing the clustering process based in Hamming distance, as suggested in [81], but implementing new algorithms for each new method is beyond the time-frame and scope of this research, so binary feature descriptors must be left out.

The only remaining descriptor implemented in OpenCV v3.1 that follows all the previous conditions is DAISY, which shall be used as a contender for the popular SIFT and SURF.

SIFT

Scale-Invariant Feature Transform was presented in [4] and is arguably the most acclaimed feature descriptor. It is based on the Difference of Gaussian (DoG) convolved over various scales of the original image, making it both scale-invariant and rotation-invariant. Although it is erroneous to generalize to all contexts, it is usually seen as the most (or at least one of the most) precise descriptor around. On the low side it can be slow. This method is patented though free to use for research purposes.

SURF

Speeded Up Robust Features [82] is viewed as the fast brother of SIFT. Instead of DoG, SURF uses difference size box filter convolved with integral image. This allows a much faster feature detection and description, though the fastness often comes with the price of a reduced accuracy. Moreover, when used with dense key points it may introduce artifacts that degrade the matching performance, accordingly to [83]. SURF retains the same invariance properties as SIFT, and it is also protected by patent but free to use for research purposes.

DAISY

DAISY [83] does not incorporate a key point detector, it was designed specifically for fast dense key points extraction. Some tests show comparable accuracy to SURF and even SIFT at a fraction of time. Similarly to SIFT, this method is based on gradient orientation histograms, the speed increase comes from replacing weighted sums by sums of convolutions, and from using a circularly symmetrical weighting kernel. DAISY is announced as scale and rotation invariant, nevertheless it usually shows lower performance dealing with it in comparison with other methods like SIFT and ORB [84].

5.2.3 Feature organization

The direct matching of features may be applied in an identification problem, but for classification it has many limitations, especially in the presence of high inter-class variability. In fact, the features detected are not complete human readable symbols, but somewhat rigid descriptions of small areas. It is not a surprise that the first attempts to match features in ancient coins had promising results [36], but still too inaccurate. Some model of organization or grouping may be helpful in this task.

A strategy that is gaining adepts is inspired on the field of natural language processing:

Bag of Features (BoF), aka Bag of Visual Words (BoVW)

In the Bag of Words (BoW) model a text is represented as a set (bag) of words. The model neglects the order of the words, or even grammar, but focus on the multiplicity of words. The concept is that a letter by itself has little discriminative value, but when grouped in words it provides a strong meaning. And the counting of words may well identify a text. No wonder this ended up being used for computer vision [85].

When adapted to images, instead of letters we have features, thus the name of the method changing to Bag of Features or Bag of Visual Words. One can cluster the feature descriptors extracted in the previous stage (using k-means for instance) and the cluster centers will act as visual words (aka features, but in a higher level sense, the nomenclature may be confusing here so is better to avoid repeating this designation), thus composing the Bag of Visual Words.

Taking a set of template images we can compile a collection of visual words and establish a reference dictionary. For each image we can now represent it by comparing each feature and its surroundings with its nearest neighbor in the dictionary and computing a histogram of the words frequency.

Recent articles with the highest successful rates stand out this strategy as a very efficient one in the context of Roman coins, especially when including spatial information [48][49][53], making it a strong candidate for this research.

5.2.4 Machine Learning

In a supervised learning approach applied to image classification, a set of images (templates) carefully labeled with its corresponding classes is trained in order to establish a model able to distinguish each class. That model will later be used to predict the class for a giving test image. The mathematical way of discriminating between classes is what distinguishes each algorithm, making them more appropriate for some scenarios and less to others. Unfortunately there is no good-for-all algorithm, so the approach in this research is to test several of the most promising methods.

A side note: in the last few years we have seen a lot of hype about Convolutional Neural Networks for its good results in many areas, even if it often needs hard-to-rationalize parametrizations. The problem is that these algorithms perform better with training sets in the order of the tens or hundreds of thousands, and up, the so-called Big-Data. As exposed in the beginning of this dissertation, Big Data is not an option in this context. Even if some work-arounds are starting to be explored, as artificially expand the "small data" for instance, this is still a very shady area, deserving its own research. Thus, even if there is no doubt that the application of Convolutional Networks on classification of ancient coins is a subject to be watchful, this approach is not tested in this research.

On the other hand, some other favorite techniques are very suitable in this context and shall be tested. It is the case of SVM, Random Forests and Naive Bayes. The older and not so promising K-Nearest neighbors is also included for comparison purposes.

SVM

Support Vector Machine [9], has been around for some time, used in many contexts, and still shows up as an excellent alternative. This discriminative classifier takes the labeled training data (the extracted features for each class) and calculates a separating hyperplane that can be used to mark new images into one class or the other, based on which side of the hyperplane they fall, thus making SVM a non-probabilistic binary linear classifier. In Figure 11 we can see a simplified example of the discriminative hyperplane calculated, the red line, H3, marks the larger minimum distance between the two classes (black dots and white dots).

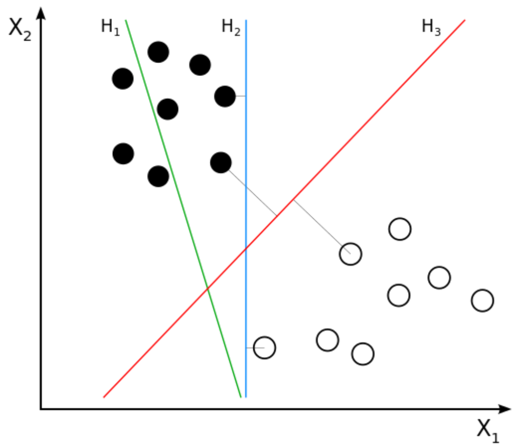

Figure 11 - Simple example of discrimination on SVM

Of course this example is an oversimplification. Besides SVM being able to deal with multi-class discrimination, the features often intersect between classes and the algorithm can perform non-linear classification mapping spaces whose dimension is higher than two.

We can have many ways of applying SVM to a classification task, these are the most common strategies:

- Multiclass ranking: SVM tries to separate all classes with a single mathematical function. This is the fastest and most compact approach, yet in many cases a single function does not exist so the approximations found may have a low accuracy.

- One-Against-All: Rather than one n-Class problem, we deal with n binary problems. One class remains on one side of the hyperplane (as a positive), all the other classes are on the other side (as negatives). In practice, this means having to learn a function for each class known. Although more computational intensive, this strategy tends to be more accurate.
- Pairwise (aka One-Against-One): Again, the big problem is reduced to multiple binary problems, but in this case a function is computed for each pair of classes, leaving us with n(n-1)/2 classifiers. As the dataset grows this strategy may become computational prohibitive for end user applications thus this strategy will be overlooked in this work.

SVM is frequently the first try (and often the last) in many classification problems. In the coin context, it is still part of the most successful methods, so it is no surprise that it is tested in this research too.

Random Forests (aka Random Trees, or Random Decision Forests)

Random Forests [86] is a method that can deal both with classification and regression, among other tasks. It is based on *Decision Tree*, a predictive model which maps binary decisions about each node (in our case it can be a feature or group of features), where each non-leaf node has two child nodes. The process leads to leaves that represent conclusions about the whole object (in our case, the class). The same class may be represented in several leaves, which represent several paths to the same conclusion. Figure 12 shows a fantasy decision tree that tries to predict the predominant color in an image. Notice that it is imperfect and very incomplete, but it is the representation the algorithm achieved based on the training data supplied.

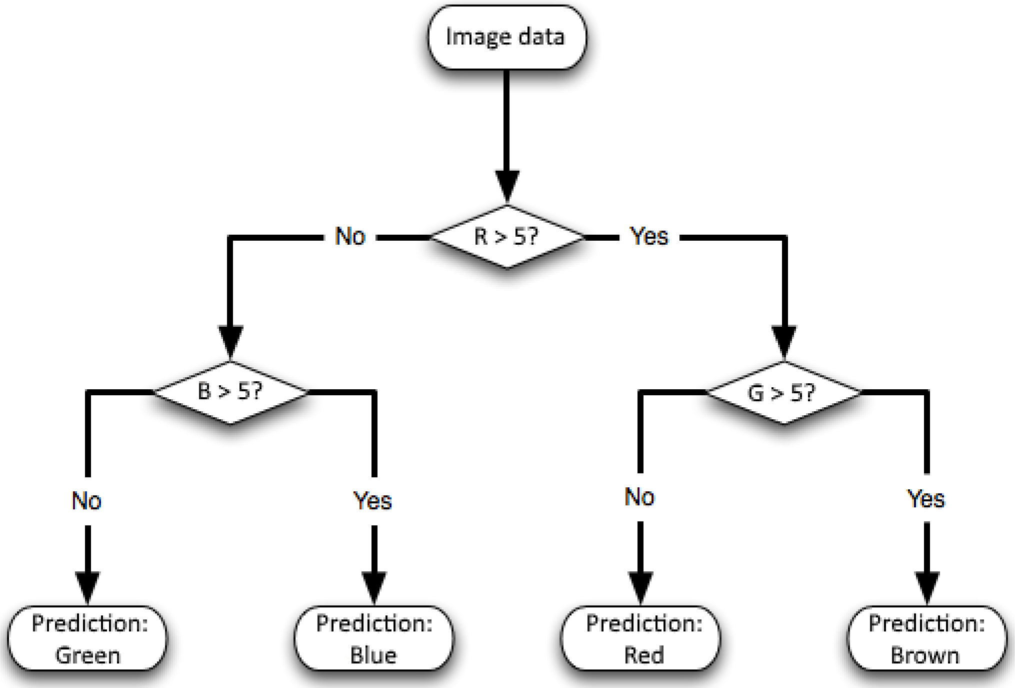

Figure 12 - A Decision Tree

A big problem about Decision Trees is that, as we allow it to grow, they tend to become over-trained, thus not being able to classify any sample we give. An intuitive explanation, using our example: as we provide more templates and allow the tree to grow, it will achieve very specific colors as navy-blue and turquoise-blue, making it more improbable to be able to identify many other tones or the ability to just answer: “It is blue”.

Random Forests overcame this problem by providing an aggregation of different Trees, each trained with the same parameters but with random sub-sets of the features. The output is the prediction that receive the majority of votes. The random sampling feeding multiple trees (formally *bootstrap aggregating)* decreases the variance of the model by preventing strongly correlated trees, thus conducting to higher accuracy.

In many problems, Random Forests tend to achieve better results than SVM, it is usually faster to train, easier to parametrize, and may provide intelligible models showing the most relevant features. It also tends to work better than SVM when there are complex interactions between features.

Naïve Bayes

Naïve Bayes [87] is in fact a whole family of simple classification techniques that represent classes as vectors of feature values, assumes they are normally distributed and usually (but not necessarily) the features values are taken as independent of the value of any other feature. Despite being considered a probabilistic classifier, it can be used without accepting the Bayesian probability. It remains a very popular method in many areas, and in some contexts is still competitive with more advanced methods as SVM, or Random Forests, thus the reason to be included in this research. A strong benefit of Naïve Bayes is that it only requires a small amount of training data, which may be interesting in this context.

OpenCV's implementation, the one tested, assumes that the continuous values associated with each class are distributed according to a Gaussian distribution. The vectors of feature values are calculated for all the training data, and for each class, a mean vector is estimated as well as covariance matrices. Those will later be used for classification of new images.

k-NN

K-Nearest neighbors is about the simplest as a classification algorithm can be. It matches the raw training data and the class of a new sample is predicted by a majority vote, i.e. the most common, of a number (k) of its nearest neighbors. To increase accuracy a weight may be assigned to each neighbor according to its distance, and that factor is taken into account during the voting. The test here presented considers this weight factor. All the computation is deferred until classification, but the process is so simple that it does not necessarily implies higher computation times compared to more advanced methods.

k-NN may hardly be expected to be as accurate as more advanced methods, nevertheless it is included in this research to serve as a base line for comparison purposes.

5.3 EXPERIMENTS AND RESULTS

5.3.1 The datasets

Two distinct datasets were used for the tests in order to better understand how their characteristics affect the classification process: a Roman coin dataset of controlled images and a medieval coin dataset of uncontrolled images.

Roman dataset (controlled images)

An image dataset of Roman republican coins, belonging to the Museum of Fine Arts in Vienna, Austria, was provided by Computer Vision Labs. This dataset was used for evaluation in [43] and it is publicly available for research purposes in [73].

The dataset is composed of a collection of coins of a very specific period and realm and, while the specimens present stylistic characteristics similar enough for the classification not to be too obvious, they also present a variety of symbols and figures that make the classification process tangible. The dataset provides 60 distinct classes with 3 different samples per class, with a total of 180 images. The images where taken in controlled lighting conditions, over a gray background, with resolutions varying between 800x800 and 2000x2000 pixels.

Figure 13 - Sample images on the Roman dataset

Figure 14 - Example of a class in the Roman dataset with its 3 different samples

Medieval dataset (uncontrolled images)

A dataset of Portuguese medieval coins was assembled during this dissertation with the aim of testing its coarser features. Although it was possible to collect more than 300 labeled images, at least two samples for each class are needed in order to be tested, and only 27 fulfill this requirement.

Nonetheless, along with the images it was requested authorization from the owners for its public research usage. Thus this dataset shall become an asset for future research, and hopefully inspire other colleagues to focus on this area.

Being medieval coins, these specimens frequently show coarser features and more degradation than the ones in the Roman dataset. Furthermore these images were taken in uncontrolled conditions, by different people, with distinct cameras, distinct resolutions, and not always the best lighting and background. As coarse as it may be, this dataset provides a more accurate example of what is to be expected from a general public use of a classification software.

Figure 15 - Example of a class in the medieval dataset with its 2 different samples

Figure 16 - Sample images on the medieval dataset

For each class, one random sample was used as test image and the remaining samples were used to compose the dictionary of the Bag of Visual Words model and as training images in all the machine learning algorithms. Although k-fold cross-validation could help stabilize the final results it was not used due to timing restrictions. Each test made took on average one to several hours (in some cases days), being the dictionary creation around 75% of the time consumed. Using this strategy would imply the creation of dictionaries for each fold, for each test made, over a hundred tests. Since the original intention was never to aim for a given success rate, but to compare the methods in order to choose the most promising, a strategic choice was made to use the time to test more variants and their combinations in order to better understand the factors that may interfere with the classification. The stability of the results was observed from the tests made with small variants and are commented on in this text.

5.3.2 The tests

The overview presented above, simplified the steps for a classification process as: Feature Extraction, Feature Grouping (BoVW) and Machine Learning (training and predicting). In detail the implementation of the method is somehow more complex and the dictionary creation constitutes a separate stage from Training and Testing:

- Dictionary Creation

  It is a one-time operation. The training images have their features extracted. Then the BoVW method will pick the features extracted from all the images and group them into a pre-configured number of clusters. Each cluster is a Visual Word, and the visual words collected establish the dictionary (also called vocabulary). This dictionary is saved into an “yml” file in order to be read and used in the other stages. Different dictionaries were built for each of the descriptors tested (SIFT, SURF and DAISY) and for each combination of variants.

  Tests were made using only the first sample of each class, and the first two samples of each class, in order to evaluate if bigger diversity translates into better dictionaries.

- Training

  Also a one-time operation. The template images (first and second samples of each class) have their features extracted and, in the BoVW approach, these are matched to the existing visual words in the dictionary. Then the visual words and corresponding labels are fed to the Machine Learning algorithm that will generate a Model which can be saved as an “xml” file to be used later.

- Testing

  Applying the prediction model to unknown images is the real goal of all the process and is supposed to be used as many times as needed. Again, the features of the image are extracted (with the same technique used in the previous stages), they are matched to the visual words of the dictionary and then, based in the model created in the training stage, the algorithm outputs a prediction of the coin class.

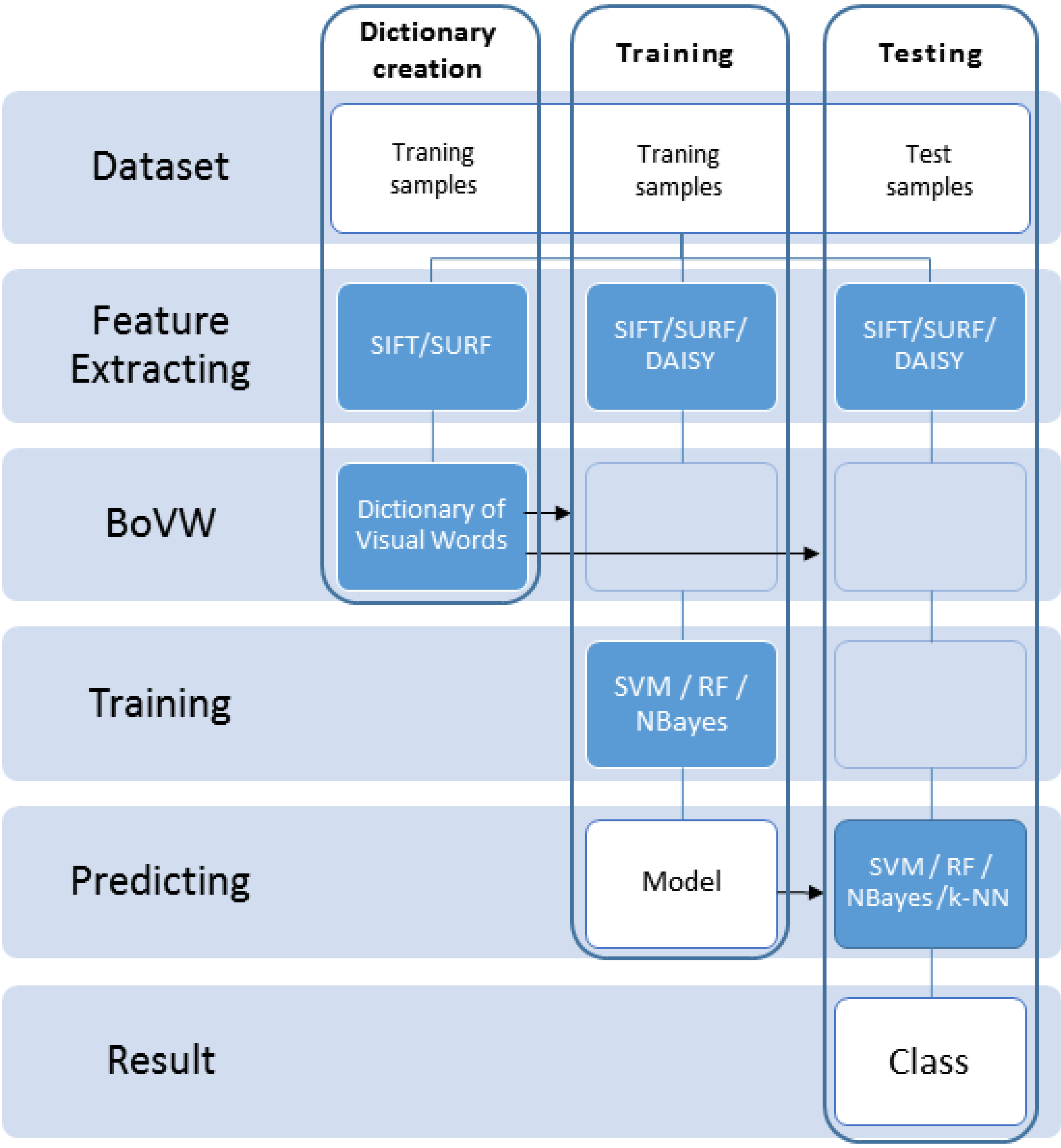

Figure 17 – Chart of the classification method

5.3.3 Results and Discussion

The following tables present a summary of the most noteworthy tests and variants that provided a step forward from the initial success rates. The percentages shown represent the correctly classified images from the test dataset.

Initial Results

The main initial tests on the Roman dataset are summarized in Table 3. As expected, SVM with the One-against-All strategy shows slightly better accuracy than SVM with multiclass, and k-NN achieves worse results than all the other algorithms. Surprisingly in this initial tests SURF provides better accuracy than SIFT, but the following more successful tests proved the opposite.

The results shown here consider the direct use of the images on the presented dataset, with a BoVW dictionary with size 200 built using only the first image of each class. The key points of each image were detected by the same descriptor used for extraction (SIFT/SURF), except for DAISY that has not detector implemented and in this case was used SIFT detector. SVM was applied with a linear kernel, since both Polynomial and Radial Basis Function always presented worse accuracy. The value of parameter C was kept very low (0.05 or below) for the best accuracy. The advisable strategy for Non-linearly separable data suggests that besides keeping C low, one should increase considerably the maximum number of iterations in order to correctly solve the problem. Nevertheless, increasing the iterations conserved or lowered the accuracy during the tests.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
SIFT	16.67%	13.33%	16.67%	15.00%	8.33%
SURF	18.33%	16.67%	15.00%	23.33%	8.33%
DAISY*(SIFT detector)*	23.33%	16.67%	16.67%	20.00%	13.33%

Table 3 - Summary of primary Machine Learning results

Pre-processing the images

Pre-processing filters were tried over all the images, always applied to all stages: dictionary creation, training and testing. Table 4 shows the results of the tests in the Roman dataset, considering the use of SIFT descriptors. A Gauss filter allows to reduce image noise. Although it shows the best accuracy improvement, the results deteriorate on Random Forests. In fact, Random Forests tends to behave worse when the images are pre-processed. Equalizing the images has similar results to Gauss filter, but at the cost of a higher computation effort, so Gauss was preferred in the following tests. Resizing the image to 480x480 pixels in order to minimize noise and minor undesired features, has some cases of worse results, but following tests proved the it tends to increase accuracy, especially when combined with a Gauss filter. Resizing also largely improves computation time. Other strategies included: normalizing the image's histogram and contrast enhancing, both showing inconsistent results among the several algorithms.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
No pre-proc.	16.67%	13.33%	16.67%	15.00%	8.33%
Histogram norm.	20.00%	13.33%	6.67%	16.67%	6.67%
Resize to 480px.	18.33%	13.33%	5.00%	18.33%	10.00%
Gauss filter	28.33%	16.67%	6.67%	21.67%	8.33%
Equalization	28.33%	16.67%	6.67%	20.00%	10.00%
Contrast enhance	23.33%	13.33%	15.00%	18.33%	8.33%
Gauss + 480px.	26.67%	16.67%	8.33%	28.33%	10.00%

Table 4 - Impact of preprocessing over the training methods with SIFT descriptors

Number of samples in the dictionary

Increasing the number of samples used in the dictionary creation (using the two images for each class instead of one, on the Roman dataset) disclosed a small improvement in most cases. Increasing the parameter *attempts* in *BOWKMeansTrainer* to the value 3 has initial variable results, but secondary tests show a general slightly better accuracy. Still these tunings do not allow to surpass the 30% success rate.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
Dictionary: 1 sample; attempts = 1					
SIFT	16.67%	13.33%	16.67%	15.00%	8.33%
SIFT+*Gauss*	28.33%	16.67%	10.00%	21.67%	8.33%
Dictionary: 2 samples; attempts = 1					
SIFT	20.00%	15.00%	10.00 %	18.33%	8.33%
SIFT+*Gauss*	20.00%	23.33%	16.67%	23.33%	5.00%
Dictionary: 2 samples; attempts = 3					
SIFT	20.00%	15.00%	10.00 %	18.33%	8.33%
SIFT+*Gauss*	25.00%	21.67%	13.33%	28.33%	8.33%

Table 5 - Machine Learning results using a bigger dictionary and 3 attempts

The dictionary size was reviewed in detail in [48], using a comparable dataset, concluding that after a certain minimum value, variations on its size result in small variations on results. They use a size of 200 as a good compromise. In our preliminary tests increasing the dictionary to 1000 presented a degradation on the results and computation times increased exponentially having taken two days to complete a Naïve Bayes test. So the size of 200 was accepted as a good value.

Dense key points

Using dense key points instead of feature detectors doubled the success rates, proving this to be a better approach in the context of coin image classification. A stride of 10 pixels was achieved as a good balance between accuracy and computation time. This means that every 10th pixel in each 10th row is chosen as a key point. Although a stride of 2 pixels may show slighter better success rates in some algorithms. Full black pixels are ignored because those (after segmentation) are background pixels so they are not meaningful, and tests have shown that the results do not vary with this condition but it decreases computation time.

For these tests, besides the Gauss filter, all images were resized by Lanczos interpolation to 480x480 pixels in order to reduce the number of key points to the most meaningful and so reduce computation effort. Additionally this resizing provided higher success rates. Table 6 shows the results for the Roman dataset.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
Stride = 10					
SIFT	51.67%	53.33%	36.67%	63.33%	28.33%
SURF	31.67%	33.33%	16.67%	36.67%	21.67%
DAISY	36.67%	28.33%	20.00%	30.00%	11.67%
Stride = 2					
SIFT	50.00%	58.33%	50.00%	58.33%	31.67%
SURF	21.67%	26.67%	21.67%	30.00%	20.00%
DAISY	33.33%	31.67%	20.00%	38.33%	10.00%

Table 6 - Summary of Machine Learning results using dense key points on the Roman dataset

Surprisingly Naïve Bayes tends to be the most successful algorithm. Also SVM with a multiclass approach shows often better results than a 1-vs-All approach. Not so surprisingly, SURF shows significantly worse accuracy than SIFT. A possible justification accordingly to [83] is that SURF may introduce artifacts that degrade the matching performance. DAISY shows variable results but comparable to SURF.

Tests on the medieval dataset

Pre-processing tests of the medieval dataset confirm that the combined use of a Gauss filter and resizing to 480x480px achieve the best result improvements. Yet, in this dataset, enhancing contrast may also increase the accuracy rates, this is corroborated by secondary tests and is particularly effective using SURF or DAISY as shown in Table 9. However when used with SIFT it tends to degrade the results for most methods, therefore contrast enhancement deserves to be taken into consideration, but shouldn't be applied indiscriminately.

Curiously SVM and Naive Bayes' results are similar in most tests. This is not a bug since the correct classified images are not always the same, but such a small 27 class dataset helps to diminish the small differences between the two methods.

Table 7 summarizes the results using the medieval dataset. These were made using dense key points with a stride of 10 pixels, using SIFT descriptors, a dictionary created with 2 samples of each class, and 3 attempts in BOWKMeansTrainer, just like the previously presented optimizations.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
No pre-proc.	25.93%	22.22%	25.93%	22.22%	7.41%
Gauss + Contrast enhance	37.04%	33.33%	14.81%	33.33%	7.41%
Gauss + Resize to 480px	48.15%	40.74%	18.52%	40.74%	7.41%
Gauss + Contrast + 480px	40.74%	40.74%	22.22%	40.74%	11.11%

Table 7 - Impact of pre-processing in the medieval dataset, with SIFT descriptors

Using a denser stride in this dataset resulted in worse success rates, demonstrating that these images have much more noise causing an excessive number of non-meaningful key points. These tests were performed after pre-processing with a Gauss filter and resizing to 480x480 pixels.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
Stride = 10					
SIFT	48.15%	40.74%	18.52%	40.74%	7.41%
Stride = 2					
SIFT	18.52%	18.52%	14.81%	18.52%	7.41%

Table 8 - Summary of Machine Learning results using dense key points on the medieval dataset

Comparing these distinct descriptors revealed an even bigger divergence between the results using SIFT and the other two methods. While SIFT results are below the ones achieved on the Roman dataset, SURF and DAISY generally dropped to half or even one third its success rates. These results are even worse if contrast enhancement is not applied, with two instances of zero successful classifications. The tests summarized in Table 9 were made using a dense approach with a stride of 10 pixels.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
Gauss + Resize to 480x480px					
SIFT	48.15%	40.74%	18.52%	40.74%	7.41%
SURF	3.70%	14.81%	0.00%	14.81%	3.70%
DAISY	7.41%	3.70%	11.11%	3.70%	0.00%
Gauss + Contrast enhancement + Resize to 480x480px					
SIFT	40.74%	40.74%	22.22%	40.74%	11.11%
SURF	11.11%	18.52%	14.81%	18.52%	3.70%
DAISY	11.11%	14.81%	14.81%	14.81%	7.41%

Table 9 - Comparison of Descriptor methods using dense key points on the medieval dataset

The tests on the medieval dataset show worse accuracy rates than the same tests on the Roman dataset. These differences were expected and may be attributed mainly to the uncontrolled images, as stated before, and to the coarseness and wornness of the medieval coins. There is also the disadvantage of having only one training sample per class instead of two. Additionally, there is one case of failure on the segmentation. Having only 27 classes, this failure may also have an impact on the results. Nonetheless, no attempts were made to improve the dataset quality since it was the original intention to simulate a general public use of the classification process.

Meaningfulness of key points

Besides the plain segmentation with the aim of removing the background, other strategies were tried in order to increase the meaningfulness of the used key points. It was noticed that the feature detectors tend to collect many points in the borders of the coins. Since these are old coins, the borders tend to be in a degraded and curved shape by it use, thus they very hardly present any distinctive feature for classification purposes (although, as previously stated, it can be distinctive for identification purposes). A more aggressive segmentation removing the extreme border of the coin (a calculated value of 5% the radius of the coin) should avoid this less representative key points and increase the success rates.

Figure 18 – Strategies to increase meaningfulness of key points for Machine Learning
(Top-down, left-right: original, segmented, borderless; edges-only; borderless and edges-only)

Another strategy tried to increase the meaningfulness of the key points on the face of the coin. Has the distinctive features on a coin are its symbols, focusing the key points only on the symbol's outlines (edges) and ignoring all the others (representing the flat portions of the coin) should theoretically provide the most relevant features and thus better results. So an extension of the segmentation was made filtering everything except the edges (with an established thickness of 5% the radius of the coin).

Contrary to expectations, these techniques have shown worse results than simple segmentation either using dense or detected key points. Removing the border at best kept the same results, but often showed a slight decline in all methods in both datasets, while removing the edges degraded even more the success rates. Table 10 summarizes the main results for both datasets. All tests shown were made using dense key points, with a previous Gauss filter and resize to 480x480 pixels. The tests over the medieval dataset include a variation using contrast enhancement.

	SVM *(1-vs-All)*	***SVM*** *(multiclass)*	***RForests***	***Nbayes***	***k-NN***
Roman dataset					
SIFT *original*	51.67%	53.33%	36.67%	63.33%	28.33%
SIFT *-border*	45.00%	51.67%	31.67%	58.33%	30.00%
SIFT *-edge*	30.00%	33.33%	18.33%	28.33%	16.67%
SIFT*-edge -bord.*	28.33%	31.67%	20.00%	28.33%	18.33%
Medieval dataset					
SIFT *original*	48.15%	40.74%	18.52%	40.74%	7.41%
SIFT *-border*	44.44%	40.74%	11.11%	40.74%	7.41%
SIFT *-edge*	29.63%	22.22%	7.41%	22.22%	3.70%
SIFT*-edge -bord.*	18.52%	26.93%	11.11%	25.93%	7.41%
Medieval dataset, with contrast enhancement					
SIFT *original*	40.74%	40.74%	22.22%	40.74%	11.11%
SIFT *-border*	37.04%	33.33%	14.81%	33.33%	7.41%
SIFT *-edge*	22.22%	22.22%	3.70%	22.22%	7.41%
SIFT*-edge -bord.*	37.04%	18.52%	11.11%	18.52%	11.11%

Table 10 - Segmentation approaches comparison

These results seem counter-intuitive. In fact stripping the image to its bear relevant symbols, as it is done by removing the extreme border and flat parts of the coins, should provide a better ground for classification and therefore better accuracy rates. However these main tests and a lot of secondary ones have shown consistently that it is not the case and results tend to degrade. This fact suggests that machine learning algorithms are considering the "empty" areas of the coin as relevant, and while as humans we tend to focus on the symbols, the truth is that the balance between "filled" and "empty" areas is also of interest and a distinctive factor between the classes. In that sense these machine learning algorithms show a correct behavior, and any attempts to remove areas of the coin in the hope of having more meaningful key points are not recommended.

Computation times

While the machine learning algorithms are extremely fast, extracting the descriptors and cluster them into Bags of Visual Words can be an intensive task. Dictionary creation takes the biggest portion of the computation times and grows exponentially as one increases its size, the dataset size, or the image resolutions. Feature extraction uses almost all the remaining total computation time, whereas creating the model and predicting a class is measured in milliseconds.

Table 11 shows the detailed times in seconds for each operation on the strategy that achieved the best general results: using dense SIFT descriptors, with pre-processed images using a Gauss filter and resizing to 480x480 pixels. The differences between the two datasets are due to the smaller number of images, as well as their lower resolution, on the medieval dataset. The operations' timings include opening the images or saving the models. The dictionary created was used for all the methods. The training and testing groups are divided into feature extraction and model creation / prediction in order to show the timings' discrepancy. The totals display the timings for processing all the images on the dataset, while "Avg. Test/image" shows in average the time it would take for an end user to receive the prediction for one image.

This table shows that computation time should not be a reason to choose between most of the machine learning algorithms, since there are no perceptible differences among them. The only exception is the training stage for Naïve Bayes: as the datasets grow, in number of images and/or their resolution, the computation time for the model creation grows exponentially. One can observe that for the medieval dataset it takes less than 5 seconds, but for the Roman dataset, with just the triple of the images, it takes almost 12 minutes. Therefore, for the not so small datasets this method can easily take several days or weeks to build a model, which may be a very serious constraint on its use.

Times in seconds	**SVM** *(1-vs-All)*	**SVM** *(multiclass)*	**RForests**	**Nbayes**	**k-NN**
Roman dataset					
Dictionary creation	2177				
Feature extraction (Train)	*686*	*688*			
Model Train		*0.221*	*3.126*	*695.266*	*0.167*
Total Training	2863	2865	2868	3560	2865
Feature extraction (Test)	*343*	*343*			
Predict		*0.120*	*0.087*	*0.336*	*0.087*
Total Test	343	343	343	344	343
Avg. Test/image	5.72	5.72	5.72	5.73	5.72
Medieval dataset					
Dictionary creation	321				
Feature extraction (Train)	*114*	*112*			
Model Train		*0.049*	*1.171*	*4.692*	*0.044*
Total Training	435	434	435	438	434
Feature extraction (Test)	*113*	*114*			
Predict		*0.042*	*0.042*	*0.090*	*0.040*
Total Test	113	114	114	114	114
Avg. Test/image	4.17	4.21	4.21	4.22	4.21

Table 11 - Computation times in the method with general best results

6

FINAL CONCLUSIONS

6.1 CONCLUSIONS

Segmentation

In general terms the new segmentation method here proposed reveals itself a very effective tool to remove complex backgrounds, with results similar or better than the current state of the art at a fraction of the time. The proposed method achieves a Good or Acceptable rate on 76% of a random collection of 100 internet images, taking an average of 0.29s per image, against 49% in 19.58s for the best found alternative method: GrabCut. The new method also reveals to be competent dealing with shadows and highlights, and excluding them from the segmentation result.

Yet there is space for future improvements. A previous method to detect the coins inside the image, using Cascade Classifiers or CNN, could allow every image to fulfil the premises for this method and allow an HB model estimation more flexible and comprehensive than the centered rectangle. Addressing images' high compression issues would also greatly increase the segmentation quality on a lot of internet images.

Although this work was focused on such a specific subject as medieval and ancient coins, the proposed segmentation method can be easily applied or adapted to other flat and uniform-colored subjects.

Feature extraction and Recognition

The first conclusion drawn from the tests made is that a dense key points approach proved to achieve better classifications than using feature detectors. In fact, success rates doubled for the better rated algorithms after applying this approach. These results suggest that feature detectors do not do a good job finding the most relevant key points on coins. Using a stride of 10 pixels provides a good balance between accuracy and computation time. In fact, using a denser stride of 2 pixels does not always translate into better results, but inevitably it requires a lot more computation time.

Further attempts to restrict the key points to the (theoretically) more relevant symbol edges also returned worse results, even if better than the feature detectors. These results suggest that machine learning algorithms are using the balance between “relevant” key points (on edges and filled areas of the coin) and “irrelevant” key points (on flat areas) in order to provide the most distinctive models. Therefore it is not recommended to try to restrict the key points either by a feature detector or by artificial methods.

SIFT steadily presented the best rates among the three descriptors tested, either using its additional feature detection capabilities, or a dense key points strategy. SURF and DAISY, although faster, never achieved the same range of success rates, therefore they can hardly present an alternative to classify ancient and medieval coins.

Concerning the choice of a machine learning algorithm, Naïve Bayes and SVM provided the best success rates, often with very similar results, especially in the medieval dataset. While Naïve Bayes achieved the single highest rate on all the tests, 63.33%, that should not dismiss the use of SVM, since the differences due to the tuning are larger than the ones found between the two methods. Additionally, Naïve Bayes can be extremely slow in the training stage of big datasets and that may be a relevant factor for an end user application.

Surprisingly, SVM-multiclass often showed better results than SVM-1vsAll. Nonetheless, the strategy 1vsAll allows the test coin to know confidence degrees for each of the classes on the dataset, i.e. when testing the image, a degree of similarity is provided for each of the classes and the highest one is chosen as the predicted class. In an end user application the chance to provide the next best predictions may be very valuable to mitigate the meaningful failure rate.

Random Forests has generally shown disappointing results. Even if in a few cases success rates are not far from SVM and Naïve Bayes, this method showed incoherent variations on the results proving to be unpredictable sensitive to variations. Even pre-processing methods as simple as a Gauss filter tend to worsen results more often that improve them when using Random Forests.

In conclusion, while these classification methods still do not provide results nearly as good as the success rates published for modern coins, the range of success rates is coming close to being usable in certain end user applications. The possibility now open to expand its use to uncontrolled images further broaden the horizons. Nevertheless, there is still ground to find new optimizations that may improve even more these results. The work started here must continue, trying other approaches and combinations, and keeping alert to new findings in the area. Hopefully, the dataset of Portuguese medieval coins collected during this work, with public permission for research use, may provide an small additional contribute.

6.2 PROSPECT FOR FUTURE WORK

The focus on this research was always to provide a basis for getting the coin recognition out of the lab to the public usage. Thus, as long as we can reach interesting results in the machine learning part, it would be interesting to really implement these methods in an end user application.

On the research point of view, a different approach to overcome the difficulties of contours detection could be the use of video in order to infer a 3D model of the coin face. With this model we would eliminate the effects of shadow and highlights and would have an accurate representation of saliencies, thus the segmentation would be a simpler task and the contours would be much more precise. Ideally this method would allow an end user to simply wave a smartphone over a coin, to get the 3D representation of its face and get a more accurate classification, or an alert on an existent identical coin (perhaps a fake or a stolen coin).

A very interesting and promising follow-up to this research would be to try to adapt traditional Convolutional Networks to the use of "small data" (instead of the more fitted concept of "big data"). Besides auspicious accuracy rates, this could provide the tools to change the focus from random feature extraction to symbol recognition, thus paving the way to style analysis (whose benefits were discussed in the introduction).

BIBLIOGRAPHY

[1] M. Zaharieva, M. Kampel, and S. Zambanini, "Image based recognition of coins – An overview of the COINS project," *Perform. Eval. Comput. Vis. 31st AAPR/OAGM Work.*, pp. 57–64, 2007.

[2] J. Jarrett, R. Hüber-Mork, S. Zambanini, and A. Felicetti, "Coinage, digitization and the world-wide web: numismatis and the COINS project," *New Technol. Mediev. Renaiss. Stud.*, vol. 3, pp. 459–489, 2011.

[3] L. Salgado and C. P. Santos, "Segmentation of Medieval and Ancient Coins Over Complex Backgrounds (a fast automatic method)," in *ICARSC*, 2016.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[5] O. Yamaguchi, K. Fukui, and K. -i. Maeda, "Face recognition using temporal image sequence," *Autom. Face Gesture Recognition, 1998. Proceedings. Third IEEE Int. Conf.*, pp. 318–323, 1998.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and Cedric Bray, "Visual categorization with bag of keypoints," *Int. Work. Stat. Learn. Comput. Vis.*, pp. 1–22, 2004.

[7] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. II, no. October, pp. 1458–1465, 2005.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2009.

[11] A. Ivakhnenko, “Polynomial theory of complex systems,” *Syst. Man Cybern. IEEE*, no. 4, pp. 364–378, 1971.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient Based Learning Applied to Document Recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] M. Nölle, H. Penz, M. Rubik, K. Mayer, I. Holländer, and R. Granec, “Dagobert – A New Coin Recognition and Sorting System,” *Proc. VIIth Digit. Image Comput. Tech. Appl.*, pp. 329–338, 2003.

[14] R. Huber, H. Ramoser, K. Mayer, H. Penz, and M. Rubik, “Classification of coins using an eigenspace approach,” *Pattern Recognit. Lett.*, vol. 26, no. 1, pp. 61–75, 2005.

[15] M. Reisert, O. Ronneberger, and H. Burkhardt, “An efficient gradient based registration technique for coin recognition,” *Proc. Muscle CIS Coin Compet. Work.*, vol. 19, p. 31, 2006.

[16] M. Reisert, O. Ronneberger, and H. Burkhardt, “A fast and reliable coin recognition system,” *Proc. 29th DAGM Conf. Pattern Recognit.*, pp. 415–424, 2007.

[17] L. J. P. Van Der Maaten and P. J. Boon, “COIN-O-MATIC : A fast system for reliable coin classification MUSCLE CIS benchmark,” *Proc. Muscle CIS Coin Compet. Work.*, 2006.

[18] L. J. P. van der Maaten and P. J. Boon, “Reliable Classification of Partially Occluded Coins,” *Proc. 2007 MUSCLE CIS Work.*, 2007.

[19] M. Zaharieva, “To SIFT or Not to SIFT? Exploring the discriminating power of SIFT features for coin classification,” *Proc. MUSCLE CIS Coin Recognit. Compet. Work.*, 2007.

[20] K. Adnan, B. Sekeroglu, and K. Dimillier, “Coin Identification Using Neural Networks,” *Proc. 5th WEAS Int. Conf. Signal Process.*, vol. 2006, pp. 88–92, 2006.

[21] T. M. Ghanem, M. N. Moustafa, and H. I. Shahein, “Gabor wavelet based automatic coin classsification,” *Icip 2009*, pp. 305–308, 2009.

[22] A. Gavrijaseva, A. Molder, O. Martens, C. Kyrkou, and T. Theocharides, “Cross-correlation-based image matching of coins,” in *2012 13th Biennial Baltic Electronics Conference*, 2012, pp. 319–322.

[23] A. Vadivelan and K. Praveen, "Implementation of Coin Recognition by Extracting the Local Texture Features with Gabor Wavelet and LBP Operator," *Int. Conf. Electr. Eng. Comput. Sci.*, no. 1, pp. 135–139, 2013.

[24] K. Wei, B. He, F. Wang, T. Zhang, and Q. Ding, "A novel method for classification of ancient coins based on image textures," *Proc. - 2nd Work. Digit. Media its Appl. Museum Heritage, DMAMH 2007*, no. 1, pp. 63–66, 2007.

[25] D. Mehta and A. Sagar, "A Survey on Various Techniques of Coin Detection and Recognition," *Int. J. Comput. Appl.*, vol. 69, no. 5, pp. 29–32, 2013.

[26] M. Kampel, R. Huber-Mörk, and M. Zaharieva, "Image-Based Retrieval and Identification of Ancient Coins," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 26–34, 2009.

[27] M. Kampel and M. Zaharieva, "Optical Recognition of Modern and Roman Coins," *Layers Perception. Proc. 35th Int. Conf. Comput. Appl. Quant. Methods Archaeol. (CAA), Berlin, 2.-6. April 2007*, p. 172, 2008.

[28] L. Van Der Maaten and E. Postma, "Towards automatic coin classification," *Proc. of the EVA-Vienna*, 2006.

[29] M. Zaharieva, R. Huber, M. Nölle, and M. Kampel, "On ancient coin classification," *8th Int. Symp. Virtual Reality, Archaeol. Cult. Herit.*, pp. 55–62, 2007.

[30] S. Zambanini and M. Kampel, "Segmentation of Ancient Coins Based on Local Entropy and Gray Value Range," *Proc. 13th Comput. Vis. Winter Work.*, 2008.

[31] S. Zambanini and M. Kampel, "Robust Automatic Segmentation of Ancient Coins," *4th Int. Conf. Comput. Vis. Theory Appl.*, pp. 273–276, 2009.

[32] S. Z.–M. KAMPEL, "Coin data acquisition for image recognition," *Proc. 36th CAA Conf.*, pp. 163–170, 2008.

[33] M. Goodman, *Numismatic Photography*. 2008.

[34] G. Hobermann, *The Art of Coins and their Photography*. H.N. Abrams, 1982.

[35] S. Zambanini, M Kampel, and M Schlapke, "On the Use of Computer Vision for Numismatic Research," *9th Int. Symp. Virtual Reality, Archaeol. Cult. Herit.*, vol. 45, no. 7, pp. 17–24, 2008.

[36] M. Kampel and M. Zaharieva, "Recognizing Ancient Coins Based on Local Features," *Adv. Vis. Comput. Part I*, pp. 11–22, 2008.

[37] O. Arandjelović, "Automatic attribution of ancient Roman imperial coins," *Comput. Vis. Pattern Recognit. (CVPR), 2010 IEEE Conf.*, pp. 1728–1734, 2010.

[38] R. Allahverdi, M. Dehshibi, A. Bastanfard, and D. Akbarzadeh, "Sasanian Coins Classification Using Discrete Cosine Transform," *Artif. Intell. Signal Process. (AISP), 16th CSI Int. Symp.*, pp. 278–282, 2012.

[39] R. Allahverdi, M. Mahdi, A. Bastanfard, and D. Akbarzadeh, "EigenCoin: Sassanid coins classification based on Bhattacharyya distance," *WCIT-2011, AWERProcedia Inf. Technol. Comput. Sci.*, vol. 1, pp. 1151–1160, 2012.

[40] S.-S. Parsa, M. Rastgarpour, and M. M. Dehshibi, "Statistical Feature Fusion for Sassanian Coin Classification," in *Recent Advances in Information and Communication Technology, Advances in Intelligent Systems and Computing*, vol. 265, Springer, 2015, pp. 75–84.

[41] A. Kavelar, S. Zambanini, M. Kampel, K. Vondrovec, and K. Siegl, "The ILAC-project: Supporting ancient coin classification by means of image analysis," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 40, no. 5W2, pp. 373–378, 2013.

[42] R. Huber-Mörk, S. Zambanini, M. Zaharieva, and M. Kampel, "Identification of ancient coins based on fusion of shape and local features," *Mach. Vis. Appl.*, vol. 22, no. 6, pp. 983–994, 2011.

[43] S. Zambanini and M. Kampel, "Coarse-to-Fine Correspondence Search for Classifying Ancient Coins," *Comput. Vis. – ACCV 2012 Work.*, vol. 7729, pp. 25–36, 2013.

[44] S. Zambanini and M. Kampel, "Using Image Analysis to Match a Coin to a Database," *Archaeol. Digit. Era – 40th Annu. Conf. Comput. Appl. Quant. Methods Archaeol.*, pp. 194–198, 2012.

[45] S. Zambanini, A. Kavelar, and M. Kampel, "Classifying Ancient Coins by Local Feature Matching and Pairwise Geometric Consistency Evaluation," *22nd Int. Conf. Pattern Recognit.*, vol. 1, 2014.

[46] S. Zambanini and M. Kampel, "A Local Image Descriptor Robust to Illumination Changes," *Image Anal. - 18th Scand. Conf.*, pp. 11–21, 2013.

[47] R. Huber-Mörk, M. Nölle, M. Rubik, M. Hödlmoser, M. Kampel, and S. Zambanini, "Automatic Coin Classification and Identification," in *Advances in Object Recognition Systems*, no. 2003, InTech, 2012.

[48] H. Anwar, S. Zambanini, and M. Kampel, "Coarse-grained ancient coin classification using image-based reverse side motif recognition," *Mach. Vis. Appl.*, vol. 26, no. 2–3, pp. 295–304, 2015.

[49] H. Anwar, S. Zambanini, M. Kampel, and K. Vondrovec, "Ancient Coin Classification Using Reverse Motif Recognition: Image-based classification of Roman Republican coins," *IEEE Signal Process. Mag.*, vol. 32, no. 4, pp. 64–74, 2015.

[50] J. Kim and V. Pavlovic, "Discovering Characteristic Landmarks on Ancient Coins using Convolutional Networks," Jun. 2015.

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *arXiv Prepr. arXiv1409.4842*, pp. 1–12, 2014.

[52] J. Kim and V. Pavlovic, "Ancient Coin Recognition Based on Spatial Coding," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 321–326.

[53] J. Kim and V. Pavlovic, "Improving Ancient Roman Coin Recognition with Alignment and Spatial Encoding," *ECCV Work.*, pp. 1–17, 2014.

[54] O. Arandjelović, "Reading Ancient Coins : Automatically Identifying Denarii Using Obverse Legend," *EECV*, pp. 317–330, 2012.

[55] A. Kavelar, S. Zambanini, and M. Kampel, "Word Detection Applied to Images of Ancient Roman Coins," *Proc. VSMM 2012 - Virtual Syst. Inf. Soc.*, pp. 577–580, 2012.

[56] A. Kavelar, S. Zambanini, and M. Kampel, "Reading Ancient Coin Legends: Object Recognition vs. OCR," *OAGM/AAPR Work.*, vol. 2013, no. 1, pp. 1–9, Apr. 2013.

[57] A. Kavelar, S. Zambanini, and M. Kampel, "Reading the legends of Roman Republican coins," *J. Comput. Cult. Herit.*, vol. 7, no. 1, pp. 1–20, Apr. 2014.

[58] Kai Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," *2011 Int. Conf. Comput. Vis.*, pp. 1457–1464, 2011.

[59] S. Zambanini, A. Kavelar, and M. Kampel, "Improving Ancient Roman Coin Classification by Fusing Exemplar-Based Classification and Legend Recognition," *Lect. Notes Comput. Sci.*, vol. 8158, pp. 149–158, 2013.

[60] S. Marchand, "IBISA 3D: Image-Based Identification/Search for Archaeology Using a Three-Dimensional Coin Model.," *Archaeol. Digit. Era – Pap. from 40th Annu. Conf. Comput. Appl. Quant. Methods Archaeol.*, pp. 186–193, 2012.

[61] S. Marchand, "IBISA : Making Image-Based Identification of Ancient Coins Robust to Lighting Conditions," *Proc. 12th EUROGRAPHICS Work. Graph. Cult. Herit.*, pp. 13–16, 2014.

[62] M. Herrmann, S. Zambanini, and M. Kampel, "Image Based Measurement of Ancient Coins," *Comput. Appl. to Archaeol.*, no. December, pp. 1–7, 2009.

[63] M. Caltabiano, G. Salamone, M. Puglisi, B. Carroccio, B. Sisalli, A. Celesti, and A. Nucita, "DIANA : an Approach to Coin Iconography According to Time and Space Through Digital Maps," *2013 Digit. Herit. Int. Congr.*, pp. 1–4, 2013.

[64] V. Kwatra, M. Han, and S. Dai, "Shadow removal for aerial imagery by information theoretic intrinsic image analysis," *2012 IEEE Int. Conf. Comput. Photogr.*, pp. 1–8, 2012.

[65] R. Guo, Q. Dai, and D. Hoiem, "Paired regions for shadow detection and removal," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2956–2967, 2013.

[66] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. S. Huang, "Total variation models for variable lighting face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1519–1524, 2006.

[67] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *Tpami*, vol. 33, no. 5, pp. 898–916, 2011.

[68] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, p. 309, 2004.

[69] P. Dollár and C. Zitnick, "Fast Edge Detection Using Structured Forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1–13, 2014.

[70] E. Ö. Faruk, K. Yılmaz, K. Lokman, and T. Ramazan, "A Vision System for Classifying Butterfly Species by using Law's Texture Energy Measures," *Int. J. Comput. Vision, Mach. Learn. Data Min.*, vol. 1, 2015.

[71] G. Takacs, V. Chandrasekhar, S. S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, “Fast computation of rotation-invariant image features by an approximate radial gradient transform,” *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2970–2982, 2013.

[72] S. Matuska, R. Hudec, and M. Benco, “The comparison of CPU time consumption for image processing algorithm in Matlab and OpenCV,” *Proc. 9th Int. Conf. ELEKTRO 2012*, pp. 75–78, 2012.

[73] Computer Vision Lab - image dataset, “Coin image dataset,” 2014. [Online]. Available: https://www.caa.tuwien.ac.at/cvl/research/cvl-databases/coin-image-dataset/. [Accessed: 17-Jul-2016].

[74] M. J. Swain and D. H. Ballard, “Indexing via Color Histograms,” in *Active Perception and Robot Vision*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 261–273.

[75] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–69, 1986.

[76] M. Fang, G. Yue, and Q. Yu, “The Study on an Application of Otsu Method in Canny Operator,” *Int. Symp. Inf. …*, vol. 2, no. 4, pp. 109–112, 2009.

[77] S. N. Tica, B. Costin-Anton, and I. Bucur, “Metallic Coin Analyzer System for Automatic Identification and Classification,” no. December, 2015.

[78] L. Gorelick and Y. Boykov, “GrabCut in One Cut,” no. Iccv, pp. 1–8, 2013.

[79] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, “ORB : an efficient alternative to SIFT or SURF,” *ICCV IEEE Comput. Soc. 2011*, pp. 2564–2571, 2011.

[80] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “KAZE Features,” in *Eur. Conf. on Computer Vision (ECCV)*, 2012.

[81] C. Grana, D. Borghesani, M. Manfredi, and R. Cucchiara, “A fast approach for integrating ORB descriptors in the bag of words model,” *IS&T/SPIE Electron. Imaging. Int. Soc. Opt. Photonics*, pp. 866709–866709, 2013.

[82] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF : Speeded Up Robust Features,” *Comput. Vis. Image Underst.*, no. 110, pp. 346–359, 2008.

[83] E. Tola, V. Lepetit, and P. Fua, “DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[84] N. Khan, B. McCane, and S. Mills, “Better than SIFT?,” *Mach. Vis. Appl.*, vol. 26, no. 6, pp. 819–836, 2015.

[85] J. Sivic and A. Zisserman, “Efficient visual search of videos cast as text retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, 2009.

[86] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[87] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2nd. ed.)*. Academic Press Professional Inc., 1990.

A

SUPPORT MATERIAL

I – ARTICLE PRESENTED TO ICARSC 2016, TO BE PUBLISHED BY IEEE

A primary approach of the segmentation method presented in chapter 4 was accepted in the 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC 2016) and gave way to a paper awaiting for publication by IEEE [3]. At the time of the conference the method had only its initial stages, meanwhile the research continued and the method has evolved to include edge information on the detection, as long as further optimizations.

The full paper, on its submitted form, is presented in the following pages (note that the paper has its own references and chapter numeration).

Segmentation of Medieval and Ancient Coins Over Complex Backgrounds

A fast automatic method

Luis Salgado
University of Minho
Braga, Portugal
lmsalgado@hotmail.com

Cristina P Santos
Microelectromechanical Systems Research Unit
(CMEMS-UMinho)
University of Minho
Guimarães, Portugal
cristina@dei.uminho.pt

***Abstract*— Coin segmentation, or separating the coin area from its background, is inevitably the first step in any robust classification method. Yet, the fact that almost every research relies exclusively on grayscale images taken in controlled environments, with uniform illumination and backgrounds, wastes a vast asset of images commonly taken by numismatists, sellers and collectors. Admittedly, very often these are not easy images to work with, combining problems ranging from high cast shadows to highly textured backgrounds. In this paper we propose a fast and automatic method to segment ancient coins over complex backgrounds using a Histogram Backprojection approach. Results are compared against a proposed automation of GrabCut algorithm, and its benefits are demonstrated. Although the present article is oriented to ancient coin segmentation, the method can also be used in other contexts presenting thin objects with uniform colors.**

Keywords-component; Segmentation; Ancient coins; Histogram Backprojection; GrabCut

I. INTRODUCTION

Coin classification has been a subject of interest for some years now, usually applied to modern coins. As technology advances, the classification of ancient coins, on its highly irregular shapes and conservations states, has come into focus too and it is getting the attention of the researchers. Segmentation, when applied to coins, usually refers to the separation of the coin from the background on an image and it is expectedly the first step in any classification method. Yet the researchers keep basing their work on little more than a couple of available image datasets, with grayscale pictures taken in carefully controlled light conditions, and contrasted backgrounds in uniform tones, in order to maximize detection results. This presents a serious problem in two ways. Firstly, we are wasting vast repositories of coin images all over the internet that could be used to make our algorithms more robust, and secondly, the same algorithms that achieve great results on these controlled images, when applied to "real world" coin images, tend to be very disappointing (the images' fault, of course).

It is undeniable that dealing with internet or uncontrolled images and their low quality standards brings great challenges. Dark cast shadows or bright reflections severely interfere with most algorithms. Poor image resolution or too high compression lead to pixilation or "block" effect, thus adulterating the shapes. The complex backgrounds are presently very hard to remove. Ancient and medieval coins provide even more difficulties, like the irregularity of its borders that very rarely are a perfect circle; some coins are so thin that they barely contrast with the background; some are so thick that their shadows extend their border on segmentation algorithms; some coins are square, some are cut in half, some have holes. To make it even worse many images display the coin inside protection holders, making it very hard for a segmentation algorithm to distinguish between the coin border and the holder border. On the upside, the computation power is now much higher than a decade ago, so there is no more reason not to use color information on our algorithms.

Although the homogeneity of the controlled datasets facilitates higher (apparent) success rates, this is a false premise. Therefore, the segmentation of the public coin images, overcoming big problems as bad illumination and terribly background choices, is the ground to more robust approaches on classification, not only narrowing the gap between researchers' and public applications, but also allowing easy access to huge image datasets.

II. RELATED WORK

The research on coin segmentation has seldom been approached in isolated studies, being rather associated to a whole classification process.

Some of the first researches with big acclamation, like Dagobert Project [1] and Coin-O-Matic [2], set the tone to the use of a massive dataset of modern coins photographed in an extremely controlled environment. In these conditions, they need only to rely on a global threshold and some basic edge detection to provide segmentation. Many researches, as in [3], frequently working over the same dataset, made use of Generalized Hough Transform (HT). The HT limitation of detecting only circles is not a problem on modern coins, so this method was broadly accepted in that context. Unfortunately ancient and medieval coins do not provide perfect circles (or no circle at all) so HT is clearly insufficient. Some pioneer ancient coins researches, as [4], proposed the use of Canny Edge, since Sobel filters provided inaccurate edge information. Even GrabCut was deprecated in favor to Canny Edge in [5], which was shown to provide better accuracy.

It 2009 Zambanini and Kampel [6] concentrated exclusively in the segmentation problem over ancient coins and came up with a method combining Local Entropy and Local Range of Gray Values. The method, still applied on grayscale images, provided good results, except when addressing the border shadows. Our tests also revealed that Local Entropy calculation tends to be very computational intensive, even more adding a simple gray range calculation according to the paper description. More recently, Huber-Mörk, Zambanini, Zaharieva, and Kampel [7], presented a coin identification method heavily based on the border shape, and thus in segmentation, and suggested the use of a connected components analysis and the same Local Range of Gray Values. The method is not detailed but seems a lot similar to [6], proving the team was very confident on its accuracy. Still, in the presence of complex backgrounds, both entropy measure and local range if gray tend to highlight the background as much as the coin, making these methods unhelpful in the present context.

All these methods were using grayscale and controlled images, at least to some degree, and surely with uniform backgrounds.

III. Proposed Method

Despite its reliefs, a coin usually presents a narrow range of colors. Even if on occasion ancient and medieval coins may present areas of oxidation or dust, usually the most dramatic color variations are on its intensity. Thus, our approach begins with isolating the coin from the background based on a probabilistic comparison of color hue and saturation. Ignoring the intensity value has the additional advantage of diminishing the shadows noise on the border detection. The next step consists in calculating a border approximation by means of a convex hull over the previous results. Fig. 1 outlines the stages of the process.

In this section we also present a comparing method based on the most widely accepted algorithm for generalized background removal, GrabCut, and propose a basic background identification to automate the process.

A. First step – Histogram Backprojection

Histogram Backprojection (HB) has been purposed by Swain and Ballard [8] back in 1992. Given a model image, the HB algorithm computes its histogram and then back-projects it in the test image's histogram in search for fitting pixels. The result is a single channel matrix with the probability of each pixel belonging to our ground model. In other words it gives us the probabilistic prediction of our object of interest.

Commonly HB is used with a previous model image like a section of skin to detect faces or hands in a test

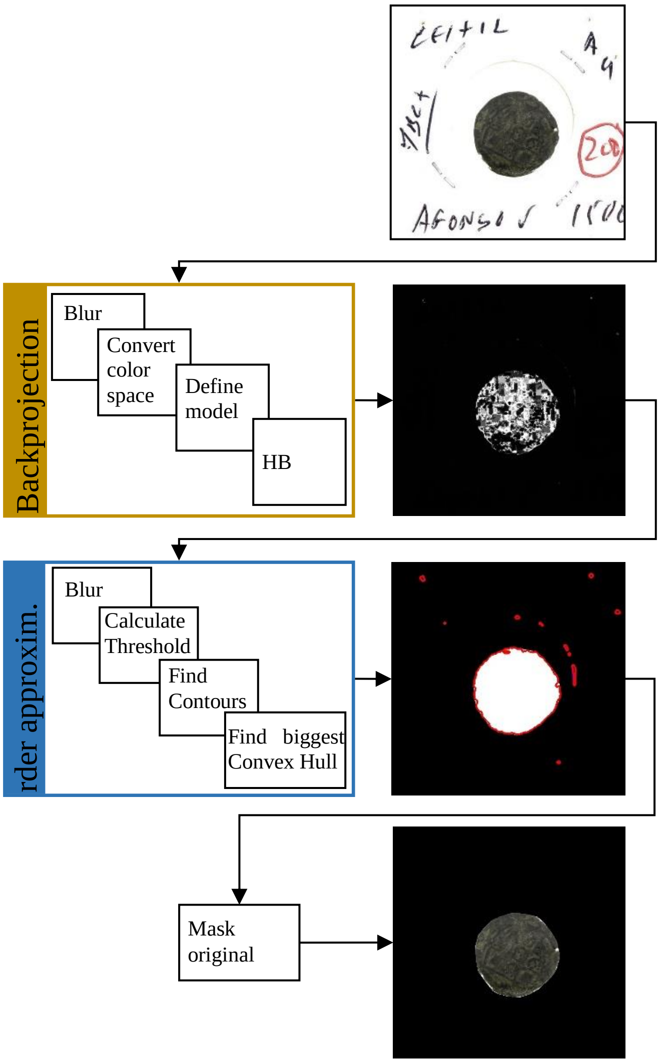

Figure 1. Proposed method. Images from top to bottom: original image; result from Histogram Backprojection; contours in red and the biggest convex hull as white area; final masked image.

image. The test image may also be used as the model to perform some kind of self-HB, assuming that the object of interest fills most of the image. Given the variability of the coins, uncontrolled images and the complexity of possible backgrounds, neither of these approaches by itself is adequate in our context.

Our tests revealed that using a partial section of the coin on our test image results on a good model for our segmentation. The larger the section model chosen, the better it covers the tone and shadow variances of the coin, thus tending to provide a more exact result. Nevertheless we cannot risk to overflow too much over the background or to choose the wrong section. In order to automatize the process, we established a rigid rectangle model with the following characteristics:

- centered in the image;
- sized on 20% of the image dimensions.

These conditions may be easily changed according to the needs of each project, yet they are not arbitrary: Even if most coins are round, using a round model would cover only its center area, thus leaving out most nuances caused by lateral lighting (shadows and highlights). The rectangular shape allows us to cover the center and to reach out a part of the outer sections of the coin with the model's corners. Also the position and dimensions of the established model force that the coin is also centered in the image and occupies at least 20% of it. Most images will easily satisfy this requirement, yet, in the cases where the coin is smaller, or there are several coins, or (the most common case) both sides of the coin are collated on the same image, we can use Haar Cascades, Local Binary Patterns (LBP), or any similar method, to identify each coin and treat them separately, one at a time (this detection step is not covered in this article).

A previous filtering with a Gaussian blur has shown to improve the results by removing noise. Gauss blur proved to be more successful than edge-preserving filters like Median blur or Meanshift, although a following addition to the image of its Laplacian may sometimes ameliorate outcomes.

We also convert the image from the usual RGB color space (Red, Green, Blue) to HSV color space (Hue, Saturation, Value or intensity). This allows us to ignore the intensity value, more prone to lighting variations, and rely solely on the hue and saturations to define our object of interest. Obviously, a grayscale image, which has only one channel based on the intensity – the channel we desire to ignore -, provides us low discriminating capabilities and it is not a good candidate for this algorithm.

The HB resulting probability matrix may be shown as an image, as in Fig. 2, and has proven very efficient in attenuating complex backgrounds. However, sometimes the border contours may seem roughly defined, thus the need to refine the result.

B. Second step – Border approximation

A Gaussian blur is applied to the output from the previous step in order to remove noise peaks, followed by a threshold to eliminate the residual noise from the background that still populates the image. At the end of these operations the background should have been completely replaced by a black level.

The automation of the process implies being able to calculate the most favorable threshold value possible. On an empirical analysis we found a relation between the ideal value and the end of the gradient from the biggest histogram peak (Fig. 3). Thus we reach an ideal value

(a) (b)

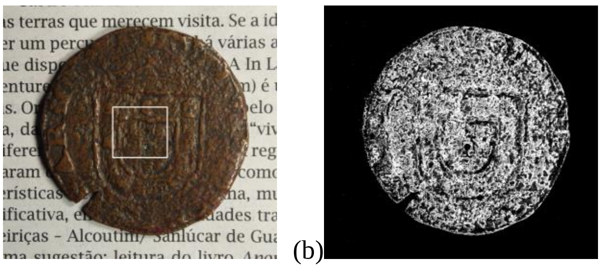

Figure 2. (a) original image with rectangle model, (b) result from HB.

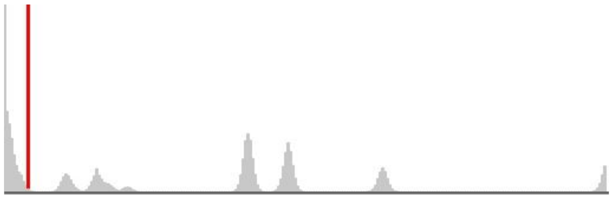

Figure 3. Histogram from HB result (flatted for display purposes) with a red line on the estimated ideal threshold value. The maximum peak usually occurs on the first bins, representing the low probability assigned to background pixels. The secondary big peaks shown here may be a symptom of a troubled result with remaining noise.

being equal to the first histogram bin, after the biggest peak, that is lower than 2.5% the maximum histogram value (i.e. if the maximum histogram value is 400, the ideal threshold value shall be the following bin number with a value lower than 10). To prevent remaining noise we also test that the next 3 pixels remain below that percentage.

By then whatever remains must belong to our coin, although it is presented as an unconnected point distribution and not a uniform shape. To find the outline we calculate the Convex Hull, which is defined as the smallest convex set that contains a set of neighbor points. If more than one shape is found, either inside the coin (from shadows or colored dirt on the coin), or outside (from remnants from the background), we choose the biggest one and take it as the outline (or border) of the coin.

As a failure recognition system, if the biggest shape's area is smaller than 20% of the image area (the size of the model rectangle) then it is marked as an assured failure and the image's threshold is set back to zero, based on the consideration that it is better to have a non-segmented image than just a small fragment of the coin. This possibility is interesting since it allows an automatic process to try again with a different method, without the need for human interaction.

We can now fill the outline calculated and use it as a mask over the original image to complete the segmentation process. In Fig. 1 we can see the whole outcomes.

C. Comparing method - GrabCut

As a comparison method we used the general purpose GrabCut algorithm [9], an approach based on optimization by graph-cut. This interactive method relies on a user defined bound box around the object of interest and estimates the color and contrast distributions from both the object and the background, isolating the connected regions inside the bound box. The process may suffer multiple iterations until the user finds a satisfactory result. Some interesting variations of this algorithm appeared meanwhile, including "GrabCut in One Cut" [10], which is interactive just the same but, as the name indicates, tries to achieve the best result in only one iteration.

In order to automate the process we choose an arbitrary bound box, 10 pixels from the image edge, on the assumption that people always leave a margin around the coin. One that wants to limit the method to round coins could also use the corners of the image.

IV. Experiments and Results

Rather than aiming for high success rates, the tests were conducted over a compiled dataset intentionally built to provide hard conditions, in the hopes of building a system as robust as possible.

A. Dataset

In order to identify the variants that interfere with the coin segmentation, the dataset of "bad cases" was constructed over 17 medieval coins, chosen to represent the combinations of their usual relevant characteristics: color, brightness and field texture. Over these coins we constructed a controlled data set contemplating hard conditions, with combinations of the following variations: lighting directions, shadow intensities," shadows over fractures (over several directions), textured backgrounds (colored coin trays, wood, marble, textured metal and newspaper) and noise (either caused by bad exposure, low resolution, or bad white balance). A small collection of 18 images of unusual specimens, like oddly shaped coins (square, oblong, holed, or concave), half coins, or unusually bright ones, was combined to the dataset. Over the refining tests the images that did not represent real variations, or repeated similar cases, were removed. In order to cover non-predicted variants, a random set of 32 coins was selected from the internet, regarding their complex backgrounds or challenges.

The resulting dataset, although partially having controlled conditions (in a harmful way), represents a collection of harsh cases that were used to study the challenges and to refine the proposed method.

B. Results

Taking into account the concerns on the dataset selection, the results shown in this paper represent the expected performance over the variables contemplated. These are not, in any way, a reflection of a real world sampling, which usually presents a much higher percentage of good images over simple white or black backgrounds, therefore easier to achieve a good segmentation.

It is also worth to note that, besides the three methods presented here, many other state of the art approaches were tested, proving very disappointing results on this specific context, and so they were excluded from this analysis.

The results are grouped according to four criteria:

- Good: when the segmentation follows closely the border of the coin.
- Acceptable: near miss, leaves a small margin or small bits of the background around the coin, but still provides an acceptable segmentation.
- Non-acceptable: near miss, similar in magnitude to *Acceptable*, but crops minor segments of the coin possibly limiting a future classification process. In other contexts both *Acceptable* and *Non-acceptable* may be joined into one acceptable/failed group.
- Failure: when the detected outline crops significant segments of the coin, or leaves large areas of the foreground.

TABLE 1. SUMMARY OF EMPIRICAL EVALUATION RESULTS

Method	**Segmentation results**				**Avg. Time (s)**
	Good	***Near miss***		***Failure***	
		Acceptable	***non-acceptable***		
GrabCut	15%	66%	4%	15%	11.018
Proposed	43%	19%	11%	26%	0.099

Some examples of the various results in each method are shown in Fig. 4 and 5.

The chosen GrabCut approach presents acceptable results, even if most fall in the class of near misses. One limitation is that, although it works well with complex backgrounds, such a large initial bound box may embrace undesired elements that will not be excluded from the result image. Furthermore this approach is very sensitive to shadows and tends to include them in its segmentation misstating the results. But the biggest deal breaker is the elevated computational times, typically around 4 seconds for very small images, and 10-50 seconds for large images, which may stand prohibitive in most applications.

The proposed method achieves a higher number of good segmentations, but still a significant number of failures, which suggests there may be room for improvements. Is interesting to note that one third of the failures are automatically recognized as so, thus allowing the automatic process to react. The biggest challenge is when the coin tones are very similar to the background. It also does not address the concave curves on the coin border, including fractures. This suggests the future improvements may embrace the inclusion of edge detection information.

This method reveals to be very efficient dealing with shadows, and excluding them from the segmentation result. Another significant advantage of the proposed method is the computation time, more than a hundred times faster than GrabCut, and allowing real time applications.

V. Conclusion and Future Work

None of the methods provide extremely high rates of accuracy in border description, so their use for cosmetic purposes is limited. Nevertheless the most usual need for coin segmentation is to be followed by a classification or identification process, and in that perspective the outcomes are very satisfactory.

The proposed method shows convincing results and proves that HB is able to give a convenient estimate of the actual coin region in a simple and fast way. Yet there is space for future improvements that may include an HB model estimation, more flexible and comprehensive than the centered rectangle; a correction for concave curves; addressing image's high compression issues; and some secondary filters in order to improve the non-acceptable cases. Also, the previous use of a detection method, like Haar cascade or LBP, would allow every image to fulfil the premises for this method.

REFERENCES

[1] M. Nölle, H. Penz, M. Rubik, K. Mayer, I. Holländer, and R. Granec, "Dagobert – A New Coin Recognition and Sorting System," Proc. VIIth Digit. Image Comput. Tech. Appl., pp. 329–338, 2003.

[2] L. J. P. Van Der Maaten and P. J. Boon, "COIN-O-MATIC : A fast system for reliable coin classification M USCLE CIS benchmark," Proc. Muscle CIS Coin Compet. Work., 2006.

[3] M. Reisert, O. Ronneberger, and H. Burkhardt, "A fast and reliable coin recognition system," Proc. 29th DAGM Conf. Pattern Recognit., pp. 415–424, 2007.

[4] M. Kampel and M. Zaharieva, "Optical Recognition of Modern and Roman Coins," Layers Perception. Proc. 35th Int. Conf. Comput. Appl. Quant. Methods Archaeol. (CAA), Berlin, 2.-6. April 2007, p. 172, 2008.

[5] J. Kim and V. Pavlovic, "Ancient Coin Recognition Based on Spatial Coding," in 2014 22nd International Conference on Pattern Recognition, 2014, pp. 321–326.

[6] S. Zambanini and M. Kampel, "Robust Automatic Segmentation of Ancient Coins," 4th Int. Conf. Comput. Vis. Theory Appl., pp. 273–276, 2009.

[7] R. Huber-Mörk, S. Zambanini, M. Zaharieva, and M. Kampel, "Identification of ancient coins based on fusion of shape and local features," Mach. Vis. Appl., vol. 22, no. 6, pp. 983–994, 2011.

[8] M. J. Swain and D. H. Ballard, "Indexing via Color Histograms," in Active Perception and Robot Vision, Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 261–273.

[9] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': interactive foreground extraction using iterated graph cuts," ACM Trans. Graph., vol. 23, no. 3, p. 309, 2004.

[10] L. Gorelick and Y. Boykov, "GrabCut in One Cut," no. Iccv, pp. 1–8, 2013.

Figure 4. Rows from top to bottom: original image; GrabCut; proposed method.
Examples of good segmentation on the proposed method. On the last column we can confirm that GrabCut has less resilience to shadows. In the same column, while globally the proposed method provides an accurate segmentation, it fails to track the small concave fracture.

(a) acceptable segmentation (b) non-acceptable segmentation (c) recognized failure (d) failure

Figure 5. Rows from top to bottom: original image; GrabCut; proposed method.
(a) Example of acceptable segmentation. Some shadow or background is shown but not relevant for classification purposes. (b) Example of non-acceptable segmentation on the proposed method. For classification use the proposed method is non-acceptable since the coin has been cropped (lower row), yet GrabCut result (middle row) is acceptable even if the error margin in segmentation is roughly the same. In other contexts both results may be joined into one acceptable/failed group.
(c) Example of recognized failure on the proposed method. The method resets the threshold and the whole image is shown. This is a curious case of *compression artifacts* caused by too high jpg compression, to which GrabCut reveals itself more forbearing.
(d) Example of failure in segmentation. Large areas of foreground are shown.

II - NOTES ON EXAMINED METHODS ON SEGMENTATION CHAPTER

According to the explained approach, the following is a light summary of the most relevant methods scrutinized from proven state of the art papers or from basic computer vision standards, and from tests conducted over the present context. Some examined methods were excluded for not following the established criteria, some provided such weak results (in this context) that dispensed further analyses, others where thoroughly explored in a detail that could not fit in such a table. These informal notes are presented here in the hope of being useful as a starting point for future investigations on the subject.

Method	Advantages / Notes	Disadvantages
<u>Segmentation</u>		
Local entropy + Local range of gray values	• outperforms adaptive thresholding • outperforms mean shift method • no parameter adjustment	• shadows still pose a problem • border tracing methods can (should?) be used to determine the exact border of the coin.
Structured Random Forests	• defines a better outline than entropy filter • faster than entropy filter, but still slow	• may not deal so well with complex backgrounds • slow (in OpenCV implementation) • results apparently worse than canny
Entropy filter		• slow • often very light, inconsistent results • useless on highly textured backgrounds
Canny edge	• outperforms GrabCut in segmentation • Increasing the contrast first (ex. With CLAHE) can enhance image details and facilitate the coin detection	• low resilience to highlights on the edge of the coin • week results in conjunction with shadow removal methods • applying Canny edge over the 3 RGB channels (and add-weighted) has similar results when applied to gray image (even worse in HSV color space) • Canny does not connect pixels into chains or segment.
GrabCut	• good results after applying constant L in Lab color space	• outperformed by Canny edge on same papers (ideal conditions) • extremely slow (4-50s) • low resilience to highlights and shadows
Gaussian Mixture Models (GMM)		• slow • low resilience to shadows • GrabCut (and similar methods) use GMM as a 'color' clustering step

		• GraphCuts is patented
Hough transform		• assumes a perfectly round coin, thus not applicable on ancient coins • very unreliable
Adaptive Thresholding		• outperformed by Local entropy + Local range of gray values
Sobel + threshold	• better and faster than Scharr • in practice is better than Laplacian in shadows and small cracks • better resilience to shadows than Laplacian • better results using "Total Gradient absolute (approximate)" than "Total Gradient (approximate)"	• theoretically should be outperformed by Canny edge • Sobel slower than Laplacian • Laplacian over 3 channels seems to have better contours (but less resilient to shadows • Sobel over the 3 channels (RGB) takes the double time and has similar results
Scharr + threshold	• better results using "Total Gradient absolute (approximate)" than "Total Gradient (approximate)"	• low resilience to light gradients • too sensible to background even after fixed Y • usually slower than Sobel and worse results
Laplacian	• very fast • contours as good as or better than Gradient • Laplacian over 3 channels much better than just over grey image • Laplacian over 3 channels is faster than Sobel; and better than Sobel for coin outline, but shows more noise.	• low contrast may difficult thresholding (over gray) • low resilience to shadows (worse than Sobel)
Gradient computing	• better outlines than Canny edge	• sacrifices shadow zones • highlights on the outer edge remain a problem
Morphological operator: Opening	• more resilience to highlights on the edge of the coin	• rounds up the bulges
OpenCV: FindCountours()		• weak results used by itself
Watershed		• unhelpful segmentation, it is hard to guess which segments should be joined to form the coin • deforms the border too much • bad results also with shadow removal methods
Mean Shift Filtering		• very slow (15 to 200s) when high spatialrad • low resilience to highlights & shadows • very susceptible to compression artifacts • no improvement for fixed Y images (shadow removal)
Saliency Detection Algorithms		• usually be slow in big images • hard to discriminate the ROI

Connected Components	• good selection of connected components • nice segmentation from background / other features	• too sensitive to shadows and highlights • inconsistent "ideal" threshold (could not find an auto estimative) • CannyEdge defines better edges • still needs border approximation (with convex hull or similar)
Image feature detection using Phase Stretch Transform		• US Patented • results no better than canny edge
Edge detection		
Sobel	• slightly better than LoG, worse than Canny edge	• uses generic operators and does not consider the image characteristics. • tends to provide inaccurate edge information in the presence of noise • theoretically should be outperformed by Canny edge
Scharr	• theoretically as fast as but more accurate than the standard Sobel	• low resilience to light gradients
Prewit		• uses generic operators and does not consider the image characteristics • worse than LoG
Laplacian of Gaussian (LoG)		• uses generic operators and does not consider the image characteristics. • low resilience to shadows • worse results than Canny
Canny edge	• seems to be the best of the basic methods	• uses generic operators and does not consider the image characteristics.
Deriche edge detector	• theoretically better than Canny edge • Deriche uses two IIR filters: one for blurring and another for derivative. As it is, IIR filter edge localization is better than Sobel filter but it slow	• slower
Roberts		• worse results than Canny Edge
ZeroCross		• similar results to LoG
Ant colony optimization (ACO)		• based on threshold, just searches for optimal threshold value • presented on modern coins, good illumination • ACO might not be the fastest algorithm available
Structured Random Forests	• capable of real time frame rates (faster than most competing state of the art methods) while achieving state of the art accuracy	• results similar to Canny edge but with double lines

Laplacian	• good results when the coins has good contrast	• bad contrast = bad results
Morphological operator: Gradient	• better outlines than Canny edge	• sacrifices shadow zones • highlights on the outer edge remain a problem
Gradient computing		• outlines sometimes worse than Canny edge sometimes better • sacrifices shadow zones
Difference of Gaussians (DoG)		• results similar to Gradient but less well defined and penalizing in the shadows • generally bad with textures
Morphological operator: Top Hat	• outline very well defined	• too penalizing in the shadows, even more than Gradient • in dark images may lose detail
Wavelet Transform (Wavelet-Based Edge Detection)		• weak results • blurs instead of lines • too variable
Shadow Removal		
Shadow removal by information theoretic intrinsic image analysis		• patented
Paired Regions for Shadow Detection and Removal		• too slow (several minutes) • needs other related images • too shallow effect
Equalize shadow		• can infatuate noise
Through RGB and HSB values pixel to pixel		• very weak results
Fix V value in HSV	• helpful in some cases	• weak results for most cases • sometimes there is severe "blocky" effect in the result (jpg losses) • does not help for GrabCut • may help for borders with Canny but not for edges
Fix V value in HSV and equalize S	• may improve color frontiers (careful with coins with dust similar to background)	• tested following by Canny and GrabCut with poor results • in general following Canny appears to have worse results than without equalization
Fix L in Lab channels	• very good results when well contrasted image • not as strong "blocky" effect as in HSV	• too low variance in many cases • bad results when similar coin and background colors are present in image
Fix Y value in YCrCb	• results similar to Lab, some better, some worse	