5-2020

The Computer-Aided Die Study (CADS): A Tool for Conducting Numismatic Die Studies with Computer Vision and Hierarchical Clustering

Zachary McCord Taylor
Trinity University, ztaylor54@gmail.com

The Computer-Aided Die Study (CADS)

A Tool for Conducting Numismatic Die Studies With Computer Vision and Hierarchical Clustering

Zachary McCord Taylor

29 April 2020

The Computer-Aided Die Study (CADS)
A Tool for Conducting Numismatic Die Studies With Computer Vision and Hierarchical Clustering

Zachary McCord Taylor

A departmental senior thesis submitted to the
Department of Computer Science at Trinity University
in partial fulfillment of the requirements for graduation
with departmental honors.

April 29, 2020

Dr. Seth Fogarty, Advisor

Dr. Nicolle Hirschfeld, Advisor

Dr. Yu Zhang, Department Chair

Dr. Sheryl Tynes, AVPAA

Abstract

Numismatic die studies are traditionally conducted by hand, and are one of the most arduous tasks a numismatist can undertake. This thesis presents the Computer-Aided Die Study (CADS), a tool that has been developed as a new way to conduct die studies using computer vision techniques. This thesis is a continuation and re-imagining of previous efforts by the American Numismatic Society to create a computational die study program, with the added intention of producing a tool numismatists can use for their research. CADS does not aim to replace the numismatist's role, but instead to aid their efforts in conducting die studies, vastly reducing pain points and time required for the process.

Acknowledgements

Thanks first and foremost is due to my thesis advisors, Dr. Seth Fogarty of Trinity University's Computer Science Department, and Dr. Nicolle Hirschfeld of Trinity University's Classics Department. Their expert counsel and constant encouragement is the sole reason this work exists. I would like to thank Dr. Peter van Alfen of the American Numismatic Society for trusting me with this project and providing his continued support throughout development. I would also like to thank Dr. Thomas Faucher of the Centre National de la Recherche Scientifique for graciously providing the data set used during development. I also extend my appreciation to Emily Herbert for helping with the LaTeX markup for many of the figures contained herein.

Finally, I offer my sincerest gratitude to my father, who seeded my interest in numismatics at an early age. I used to ask the Tooth Fairy for old coins - he managed.

Contents

1 Introduction

Numismatics is the study of coins and coin-like objects. Numismatists study many facets of coins spanning a broad range of topics - iconography, metal content, weight, methods of manufacture - but this work is concerned with what coins tell numismatists about what's missing: *the tools that made them.*

Coin manufacture has been carried out by hand for much of history, with some of the first coins originating in Ephesus in the mid to late 7^{th} century BCE [9]. *Dies*, the tools of coin manufacture, were hand cut by engravers, and thus no two dies were identical. Numismatists leverage these slight variations through the *die study.* Because no two dies are the same, yet a single die produced many coins, coins produced by the same die can be identified and grouped together through careful study [1].

Die studies are traditionally conducted by hand, and are one of the most arduous tasks a numismatist can undertake. This thesis presents the Computer-Aided Die Study (CADS), a tool that has been developed as a new way to conduct die studies using computer vision technology.

Die studies are highly suitable for automation due to their repetitious nature, requiring constant comparisons, sorting, and clustering of coins. The increased digitization of numismatic data sets in recent years make die studies further suited to a computational approach [10]. Perhaps the most significant impetus for seeking a computational die study is that the scale of coin issues left to be studied, such as the late 5^{th} century BCE Athenian tetradrachms, far outweigh the abilities of numismatists by traditional methods.

The American Numismatic Society has been pursuing a computational die study for quite some time [47]. This thesis is a continuation and re-imagining of those efforts, with the intention of providing a complete tool numismatists

can use in their die studies. CADS focuses on the "aided" part of its name - it does not aim to replace the numismatist's role, but instead to aid their efforts in conducting die studies, vastly reducing pain points and time required for the process.

A survey of previous applications of computer vision in numismatics finds a single category of work: classification [26, 39]. Much work has been done in the classification of modern coinage into its appropriate denomination, especially where less-sophisticated methods fail [39]. For example, many small-denomination Indian rupee coins are approximately the same size and composition, so normal sorting methods based on coin size and metal content cannot be used in devices such as vending machines. In these situations, more sophisticated methods such as image recognition are attractive solutions, and thus a great deal of work has been accomplished in classifying modern Indian coinage quickly and with a high degree of accuracy [48, 45, 35].

Modern coinage struck with precision cut dies is well-suited to classification with computer vision techniques. The same is not so easily said about ancient coinage struck by hand-carved dies, as the task of classification becomes more difficult when inter-class variance is as high as it is in ancient issues [2, 26]. As a result, the body of work bridging computer vision and ancient numismatics is small. Die studies are concerned with sub-classification, which fortunately does not have the same pitfalls as faced when classifying ancient coins.

This thesis presents the CADS tool: an alternative, computational, method for accomplishing the same tasks in far less time. Chapter 2 presents an overview of the methodology and significance of traditional die studies. Chapter 3 discusses the computer vision employed by the CADS tool. Chapter 4 discusses the methods employed by CADS, outlining what steps of the process are automated, what parts are computer-aided, and how the tool can be used to aid

a die study. Chapter 5 presents preliminary results from the current state of development of the tool, and Chapter 6 suggests some directions of future work.

2 Numismatics

Before discussing the computational aspects of CADS, it is necessary to present some background on the numismatic topics that make up the foundation of this work. This section introduces some numismatic terms through a brief overview of the minting process, then discusses die studies in detail, highlighting their methods, challenges, and unique significance to numismatics and the wider body of classical studies.

2.1 Minting Process

We have a good understanding of how ancient coins were typically manufactured, partly due to the fact that the process has not changed much since its inception in the mid to late 7$^{\text{th}}$ century BCE [9, 41, 24]. In general the process involved five basic components [24, 32, p. 11-17]:

1. A *blank*[1] of a precious or semi-precious metal
2. An *obverse die* (See Figure 2) of bronze or iron
3. A *reverse die* or *punch* of bronze or iron
4. A hammer or other striking implement
5. An anvil or other sturdy base

To manufacture a coin, the obverse die is set in the anvil with the blank set into the obverse die. The reverse die or punch is then positioned over the blank in line with the obverse die below. The reverse die is then struck with a hammer to impart the designs of the dies onto either side of the coin [24]. Figure 2.1 illustrates this process. This action of striking the reverse die is often used to

[1] Also called a *flan* or *planchet*.

refer to the entire manufacturing process - it is not uncommon to refer to a coin as having been "struck" on a certain date, or by a certain mint, etc.[2]

While the manufacturing method described above is representative of mint practices for the majority of antiquity, there were several earlier iterations of coin manufacture [32, p. 17]. The most common deviation is the use of a punch to force the metal of the flan into the obverse die. Rather than a reverse design imparted by a hand-carved die, these early coinages have a simpler pattern imparted on the reverse by the blow of the punch, referred to as incuse punches. Over time incuse punches became more intricately decorated, paving the way for full-blown iconography with dedicated reverse dies [32, p. 17].

2.2 Dies

Very few coin dies have survived into the present. There are likely as few as a couple dozen authentic surviving examples [49]. This is likely because ancient mints had to combat counterfeiting, and thus kept close watch over their dies. Scholars believe that dies were often recycled for reuse, or destroyed completely when exhausted or broken during use [49]. Even if some dies did make it outside an ancient mint, it is likely they were melted for their iron or bronze, as such materials would have been highly valuable. What can be learned from the few surviving ancient coin dies is limited in scope, as they are either severely rusted, have poor provenience, or are ancient forgeries [49]. Figure 2 depicts a 16^{th} century reproduction of a Roman coin die, and it is likely close to what one would have looked like in antiquity.

[2]The "strike" of a coin can also refer to the degree to which the details of the coin are pronounced, derived from a harder strike in the minting process doing a better job of imparting the die imagery to the coin blank.

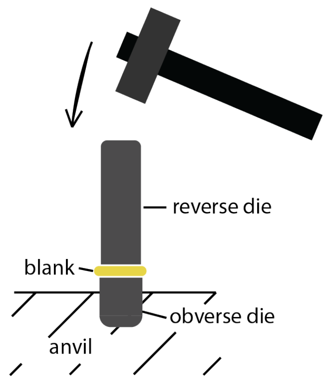

Figure 1: Diagram of minting process [33]

Figure 2: Reproduction of a Roman coin die [34]

2.3 Die Studies

Consequently, numismatists study dies indirectly through the coins they have struck [1, 7]. This takes the form of a *die study*, in which coins of the same type are grouped with coins struck by the same die. Coin dies were hand-carved by a mint worker (called the *celator* [32]), resulting in no two coin dies being identical. This is what makes the die study possible: slight variation between hand carved dies that researchers can use to group coins together [1].

Die studies typically involve an initial data collection phase, where information is gathered on the issue being studied. Often times die studies will pool their data from a mixture of sources, such as museum collections, public databases, auction listings, hoard publications, and more. Physical coins can be studied if access is possible, but images are more common and allow for coins from various sources to be included in the study without having physical access.

Once data is collected, the coins need to be compared to one another to identify potential die links. Herein lies the most tedious step — this process is done by hand, one-by-one, and with great care [7]. The difficulty of this task is highlighted in Figure 3, which depicts two coins struck by the same die. One method is to print out each coin image and paste them onto index cards, sorting alike coins into piles for further refinement, until individual die links can be determined.[3]

Once die links are determined for both the obverses and reverses of the coins the die study can be considered complete, but many numismatists take this further and look for evidence of things like die chronology and *hands*.[5] Die

[3] It is also important to note that these die links occur on both the obverse and reverse sides of the coins, but more frequently on the obverse. This is due to reverse dies breaking more often than the obverse dies, as reverse dies took the blow of the hammer while obverse dies were set in an anvil.

[4] Ptolemaic tetradrachm images provided courtesy of Dr. Thomas Faucher of the Centre National de la Recherche Scientifique, and Dr. Julien Olivier of the Bibliothèque nationale de France, who graciously provided data from their Paphos hoard study for this work [19].

[5] A *hand* is the work of a single celator spanning several dies.

Figure 3: Two Ptolemaic tetradrachms struck from the same die[4]

chronology is possible to determine because obverse and reverse dies experienced wear at uneven rates. Because reverse dies were subjected to the blow of the hammer, they wore out much faster than the obverse dies, which were set in an anvil and thus more resistant to wear [20, 1]. This made it unlikely for a pair of dies to wear out at the same time.

The typical notation for die chronology uses capital letters for obverse dies, and lowercase letters for reverse dies [1]. Consider a coin of the form A-a. Since reverse dies wear out more often than obverse dies, it is often the case that within a study there will also exist coins in the form A-b, A-c, and so on. This sequence of a single obverse die with multiple reverse dies is common. The ordering of reverse dies can be determined by examining the state of the obverse die - later reverses will have obverse strikes with higher degrees of die wear [7].

Obverse dies break as well. If the study is large enough to contain multiple reverse die matches,[6] then it is possible to continue the A-a, A-b, A-c sequence introduced above by finding a reverse die with a new obverse, denoted in this example as c-B. Figure 4 illustrates this sequence. Another intuitive notation commonly encountered is depicted in Figure 5, where the chronological flow of time in the downwards direction is more clear. Coin A-a is struck first in this sequence, with coin C-d being struck last. Die links are explicitly noted. Identifying chronologies in this way is a useful method for numismatists to gain

[6]The population of reverse die links is often far smaller than that of obverse dies, making the linking of obverse dies by common reverse die less common than the other way around. Examples of die counts can be found in [19, 46].

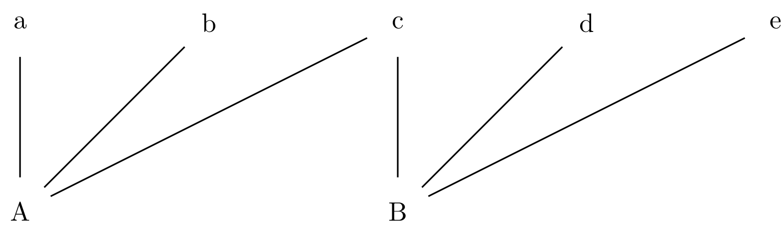

Figure 4: An example of die chronology

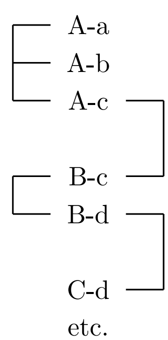

Figure 5: Alternate notation for die chronology

a better understanding of die usage within a single coin issue.

2.3.1 Die Study Process

For the purposes of this work it will be helpful to formally define the steps of a die study process as described in the previous section. This will make it easier to highlight which parts of a die study CADS has been developed to automate, and which have been left to the researcher.

In general, a die study consists of the following steps:

1. Gather data
2. Clean up images
3. Identify key comparators
4. Sort images into rough piles, refine
5. Identify and confirm die links
6. Further analysis (chronology, hands, etc.)

This process encompasses most of what a die study entails [7]. Data is collected from widespread sources, images are prepared for comparison, key regions of the iconography are chosen as points of interest for comparison, then images are painstakingly sorted through to decide which are similar enough to have been struck by the same tool. Only then may the researcher move onto the more interesting analysis.

2.3.2 Challenges of Die Studies

Die studies pose significant challenges for numismatists [47, 1, 7, 21]. Coins are subject to circulation, variation, post-mint damage such as test cuts (see Figure 6), and other things that can make determining die links exceedingly difficult. To make matters worse, die studies are often conducted on a large number of coins, often in the several hundred to thousands [21, 19, 46].

Figure 6: A test cut in an Athenian tetradrachm

As a result, conducting a die study demands a significant time commitment, as each coin needs to be compared to each other coin at least once to determine visual similarity. This relationship is quadratic to the number of coins in the study *in the best case*, where the numismatist is hypothetically capable of determining a die link with only a single comparison. Thus, for a die study of 3000 coins, determining die links for obverse dies alone would take 3000^2 or $9,000,000$ manual comparisons.

It is this feature of die studies that has kept certain issues out of reach as topics of study. Wolfgang Fischer-Bossert's die study of ca. 8000 didrachms of Terentum in Southern Italy "cost [him] nearly a decade of his life and a good deal of his eyesight to complete" [21, 47], yet some issues have many times that number of surviving examples. The Athenian tetradrachms depicted throughout this work are estimated to have roughly 60,000 surviving examples [47].

The words of M. H. Crawford sum up the issue: "the practical problem is that counting all the dies used to strike during the Republic would be the work of several lifetimes" [12]. Indeed, if it took Fischer-Bossert the better part of a decade to complete a study of around 8000 coins, numismatists have no hope of tackling larger issues by traditional methods [21, 47]. There are also coin issues

where the population is so low that the die count is close to the population of known coins in that issue, placing a lower bound on what issues are available for numismatists to conduct die studies [1].

These challenges come together to make die studies perhaps one of the most tedious and painstaking undertakings in numismatics [1, 47]. On top of the difficulties within a particular study, there are a limited number of coin issues that can even be considered for a die study, as many either have too few surviving examples or are so large that the die study would exceed what is possible in a researcher's lifetime [1].

2.3.3 Significance of Die Studies

Die studies find their significance in what they add to historical context outside of the individual coins on which they focus [7, 1]. At a glance, die studies generate statistics that help scholars perform certain comparative analyses of the city or city state from which the coins originate [1].

In general, there are two major benefits to a die study:

- Die studies can provide a better understanding of production (methods, scale, relative chronology, etc.) within a particular series, and
- Beyond a particular series, the totality of die studies from a city or city states in a region can allow for comparative analysis of economic factors.

While die studies are extremely useful in gaining a better understanding of a particular series, the latter benefit appeals to a wider audience. Aside from their iconography alone, it is through this aspect of die studies that coins contribute to the wider body of historical research.

Once a die study has been completed, the number of observed dies can be input into statistical estimation models, such as those devised by Warren Esty

[18], to get an approximate number of dies used to strike the entire issue. Once the number of dies used in an issue is approximated, it can be multiplied by the number of coins a die could have produced to get an approximate count for the entire issue in antiquity.[7] This number can then be used for comparative analysis, based on the amount of raw precious metals (gold and silver) needed to strike the issue. In this way, die studies provide historians a window into the comparative sizes of ancient economies, the scale of their coin issues, the extent of trade for the issuance of coinage, and more.

The impact of the challenges detailed in the previous section are now abundantly clear: die studies can provide valuable insight into ancient economies, yet studies on many ancient Greek coinages such as the 5$^{\text{th}}$ century Athenian tetradrachms are out of reach for traditional die studies [47].

3 Computer Vision

Computer vision deals with enabling a computer program to ingest, understand, and manipulate visual input such as images and videos, and consists of a vast encyclopedia of tools. This work will refer exclusively to computer vision as it applies to image processing, as video data is not relevant to CADS. Most computer vision workflows comprise of input, feature extraction, and some sort of feature analysis. More complicated analyses might include additional layers that incorporate machine learning methods or other advanced techniques, but in the general case a system ingests visual data and processes it in some way.

In CADS, computer vision is broken down into three components: data ingest, feature detection, and feature analysis. This chapter will give an overview of what each component entails, and its relevance to CADS.

[7] A common multiplier is 20,000 coins per die, but this number is debated [13]. See [20] for experimental efforts to attain a more accurate measure.

3.1 Data Ingest

Data ingest typically consists of converting images to standard formats and sizes, correcting any rotational or scaling issues, and performing some pre-processing to the images to prepare them for further analysis. The pre-processing techniques of interest for this work are grayscale conversion and blurring.

The first pre-processing CADS utilizes is grayscale conversion, which takes the 24-bit RGB color values in each pixel of an image and converts them to an 8-bit grayscale luminance value [38]. Figure 7 illustrates what a coin image looks like before and after the grayscale conversion process. A gold stater has been used to emphasize the color change that would otherwise be too subtle to notice on a silver coin.[8]

While such a conversion might seem a strange thing to do for human eyes, it makes sense for images that are going to be processed by a computer. Perhaps the most significant benefit of pre-processing images in this way is the reduction in the dimensionality of the underlying image data. Many computer vision algorithms are only concerned with changes in contrast or significant identifiable shapes and features of an image, which can have nothing to do with color. In the case that they do, grayscale still preserves the relative differences between lighting and color intensity in different regions of the image [38]. Further, more advanced computer vision techniques such as convolutions are more easily applied to two-dimensional grayscale data, and are orders of magnitude more complex to deal with colored images, for marginal added benefit.

The other pre-processing step CADS takes is blurring: a noise-reduction method used to increase efficiency and reduce susceptibility to inconsistencies in image data [25]. Figure 8 depicts the effects of median blurring. Common methods include the Gaussian and median blur, which both result in a "car-

[8]It is also worth noting that any point made by visual example of grayscale conversion will be lost in black-and-white print versions of this work, but at least there was an attempt.

(a) Full color image[9]

(b) Converted to grayscale

Figure 7: The effects of grayscale conversion

toonish" image that still resembles the source image, but has had much of the detail unnecessary to feature extraction smoothed out. The resulting image still has the major features of the source image, but detail that is not necessary for feature extraction is removed and thus computational complexity is further reduced. When used together, these techniques prepare images for the next step in the computer vision pipeline.

(a) Without median blurring

(b) With median blurring

Figure 8: The effects of median blurring

3.2 Feature Extraction

After images have been pre-processed, computer vision algorithms extract key features and descriptors that can be used to programmatically understand and evaluate each image further down the computer vision workflow [22]. Intuitively, a feature is an identifiable component: the stem of an apple, someone's left eye,

[9] ANS 1944.100.406, American Numismatic Society, accessed April 25, 2020, `http://numismatics.org/collection/1944.100.406`.

the corner of the building, and the like — anything that can be used to match similar features in other images. In the context of computer vision, a feature refers to an identifiable point or region of an image that can be described and then matched — with a reasonable degree of confidence — to an occurrence of the same feature in another image. Feature extraction is done automatically by standard algorithms that accept an image as input and provide descriptive features as output.n example

A critical attribute of features, which allows them to be matched to features on other images, is *invariance*. Invariance refers to a feature's resilience to changes in scale, rotation, and translation. When matching features in two different images, invariance permits features to be matched even if they appear in different orientations.

CADS employs the Oriented FAST and rotated BRIEF (ORB) feature extraction algorithm [36]. An example of features detected by the ORB algorithm is depicted in Figure 9, where colored circles indicate the location of each feature. ORB is a faster and more efficient alternative to the more common Scale-invariant feature transform (SIFT). Both algorithms do the same thing: they extract rotationally and scale invariant features from images that can be matched to features in other images processed in the same way. The ORB algorithm is slightly more sensitive to changes in scale, but this is an acceptable trade-off for the increased efficiency [42]. Invariance is key here - were features extracted by ORB and SIFT algorithms not rotationally and scale invariant, it would be virtually impossible to compare any two images that were not taken in a controlled environment under identical photographing conditions.

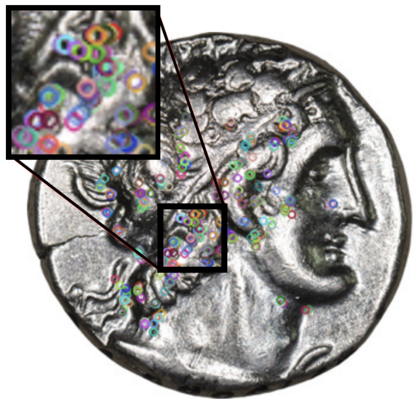

Figure 9: ORB features extracted from a Ptolemaic tetradrachm

Rotational and scale invariant feature extraction, when used in conjunction with other pre-processing techniques, allows images taken in different lighting conditions, of various sizes and scales, and at different rotational angles, to be compared efficiently and effectively in the next step of the computer vision workflow.

3.3 Feature Analysis

Feature analysis takes the features extracted from an image and extracts high-level information, for instance by matching features across images; comparing the features to a corpus with known labels; or running them through a machine learning algorithm. As feature analysis is the last step in the workflow, the information gained from the images should be high-level, so that it is usable by other programs or users interacting with the data produced.

This section will introduce basic feature matching, as well as provide a brief overview of some common types of feature analysis used in computer vision, with a heavy focus on sub-classification, as it is highly relevant to this work.

3.3.1 Feature Matching

Feature matching is the process of matching features observed in one image to occurrences of the same features in another image. This requires a notion of *distance* between two features, which is a measure of similarity that is obtained by comparing two features' underlying bit vectors. Implementations of distance functions vary, but their output can all be interpreted in the same way: the smaller the distance between two features, the more likely they are to be a match.

A simple method of matching features is by brute force, where each feature in one image is compared to each feature in the other image. Although this requires a quadratic number of comparisons, features are typically represented as bit vectors [36], making these quick to perform. Figure 10 depicts the result of brute force feature matching, with colored lines drawn between each feature match.

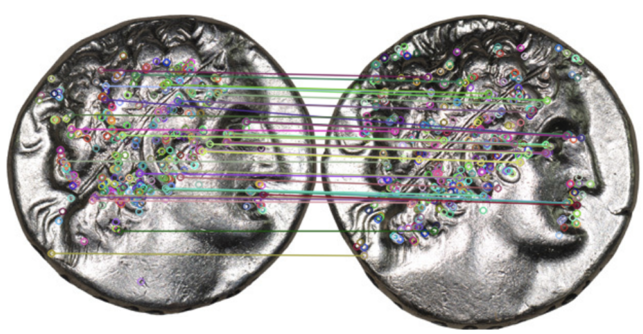

Figure 10: Feature matches between two Ptolemaic tetradrachms

In the above figure, the pair of images displays the 40 feature matches with the lowest distance scores. These distance scores can be used to compare regions within an image, to determine the similarity between two images, to cluster similar images together, or in other feature analysis techniques.

3.3.2 Feature Analysis Techniques

Feature analysis techniques can be split into two categories: *supervised* and *unsupervised* [40]. Whether or not a technique is supervised depends on its data requirements. Supervised techniques use a corpus of labeled training data. New, unlabeled, inputs are compared to the existing data to determine the most likely label. By contrast, unsupervised techniques do not require already-labeled training data, and instead search for similarities in the inputs [40]. Supervised techniques are typically concerned with classification, and unsupervised techniques deal with sub-classification and clustering. This section will discuss both classification and sub-classification.

3.3.2.1 Classification The task of a classifier is to determine a label for an image given its extracted features. This could be as simple as identifying the numerical value of a hand-drawn digit, or as complex as generating labels for individual items within an image. Figure 11 depicts an example of labels generated for a coin image by Google Vision AI API, a state-of-the-art classifier.

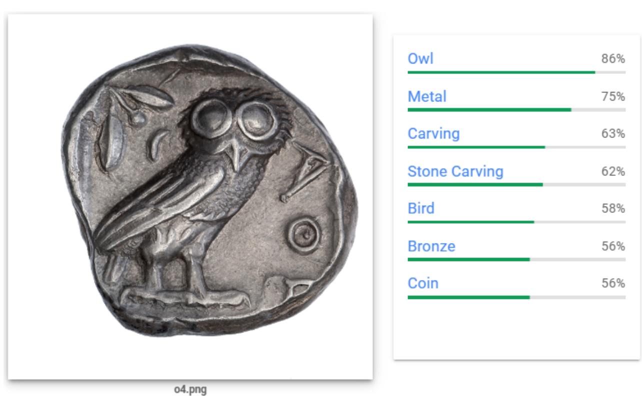

Figure 11: Labels generated from an image by a classifier

Note that the labels have a percentage score - this is the *confidence* the classifier has assigned to the label. It indicates how confident the classifier is

that the given label is a correct match.

Common classification methods include neural networks, Bayesian classifiers, decision trees, and more. They find their strength in being able to achieve high accuracy in classification, but they require vast quantities of high-quality labeled training data to be successful.

3.3.2.2 Sub-classification The task of a sub-classifier is to determine groupings within a class. Clustering algorithms are typically unsupervised and can determine inter-class groupings automatically without previously labeled training data [37, 50, 30, 27]. This makes them especially useful in situations where there is a lack of good training examples or the proper grouping is unknown.

This work is exclusively concerned with sub-classification, as die studies are typically performed on a single type[10] or very closely related types of coin, and thus the class is already known. Further, the methods CADS uses for sub-classification do not require prior-labeled training data, and thus can be used on poorly understood and/or low population types.

While classification relies on prior-labeled training data to match new images to a class label, sub-classification takes images within a known class and attempts to divide them further based on some distance function. This is particularly relevant to die studies, as they are exactly a form of sub-classification — the type of coin is already known, it is the individual groups of dies *within the type* that the researchers wish to ascertain. This can be done with clustering algorithms that repeatedly compare images to one another and determine which images are the most alike. Features extracted from images can be input into a distance function to determine how semantically far two images are from one another. Images with many similar features are semantically considered to be closer by the distance function, represented by a smaller numerical output.

[10] A coin's *type* refers to its unique iconography.

Clustering algorithms group together images with small distances to one another, in such a way that each individual cluster of images has a low average distance among its members. A common example of one such clustering algorithm is the k–means clustering algorithm [27]. Other unsupervised clustering methods include hierarchical clustering, which merges images using the same distance function but preserves the order in which they were merged, producing a hierarchical structure that encodes ever-increasing cohesion from the root of the hierarchy to the leaves [30, 31].

4 Methods

CADS is designed to aid numismatists in conducting die studies. It includes an automated clustering component, which uses computer vision to create a dendogram: a visual organization of the coins by similarity. It then contains an interface to allow users to interact with the dengrogram and identify dies. This section reviews the technology stack used in the CADS tool, and discusses the the computer vision workflow used to determine die links.

The computer vision process in CADS is automated, other than two components that require input from the user: the feature radius and blurring radius. Each step of the process takes input from the previous step and produces an output that it passes to the next step down the line. Once the process is complete and an output has been produced, CADS turns control back over to the user, and its computer vision work is complete.

4.1 Technology Stack

As CADS was developed with plans to be released as a fully fledged application, a great deal of planning went into its technology stack. The authors were charged with a few loose requirements by the ANS that helped guide the process:

CADS should...

- Be usable by non-technical researchers
- Work with both Mac and Windows devices
- Analyze coin images to help users identify potential die links

These three requirements led CADS to its current state: a cross-platform desktop application with a graphical user interface (GUI), running computer vision algorithms on the back-end. This section details these components of CADS - the frameworks involved in building it into a cross-platform application, the user-interface design, and the computer vision algorithms and workflow used by the tool to help researchers identify potential die links.

Open-source frameworks were used where possible for speed and ease of development. This placed development focus on the computer vision workflow, minimizing time spent wrestling with user interface design.[11]

To make CADS a cross-platform desktop application, the Electron.js framework was used [14]. Electron allowed CADS to be developed much like a web application using Node.js and other web frameworks, while being compatable with both Mac and Windows operating systems as a standalone desktop application.

User interface design in CADS follows Google's Material Design Guidelines [23] for usability and ease of design, as pre-built components are available for use in web development, which were easily adapted for use in CADS.

The main dendrogram display in CADS uses d3.js [6], and the core visualization logic is heavily adapted from a `bl.ocks.org` post by user d3noob.[12]

[11] Nevertheless, much wrestling occurred - albeit orders of magnitude less than possible.
[12] `https://bl.ocks.org/d3noob/43a860bc0024792f8803bba8ca0d5ecd`

For the computer vision workflow CADS heavily utilizes OpenCV [8] through a Node.js wrapper, opencv4nodejs.[13] The Node.js wrapper allows JavaScript code running in Electron to call native OpenCV binaries compiled on the user's machine. This is convenient as it provides far more computational power than Electron has on its own through its embedded Chromium browser, where the CADS UI is running.

4.2 Data Ingest

The first step in the CADS computer vision workflow is data ingest. This process is initiated by the user, giving CADS the file system directory where the coin images (the *dataset*) reside on the user's computer. CADS must then read each image into memory and perform some light pre-processing as introduced in Section 3.1 for efficiency and accuracy.

Each image is read into an OpenCV Mat, the base data structure for all of OpenCV's image analysis capabilities [8]. Once an image is read into a Mat, the image is converted to grayscale as introduced in Section 3.1 [38].[14] Median blurring is then applied to the image as introduced in Section 3.1, which reduces noise and helps feature extraction focus on the most visually interesting parts of the image [25]. Figure 12 depicts this workflow in CADS. Note that Figure 12c has had most of the circulation wear from Figure 12b blurred away, and thus such wear can be ignored by CADS.

After these two effects have been applied to each image loaded into memory from the dataset, CADS automatically moves onto feature extraction.

[13] `https://www.npmjs.com/package/opencv4nodejs`

[14] opencv4nodejs provides a convenience method $Mat :: bgrToGray$, which was used. In C++, the $Mat :: cvtColor$ would be required.

(a) Original (b) Grayscale (c) Median blur

Figure 12: The CADS data ingest workflow

4.3 Feature Extraction

CADS uses ORB features as introduced in Section 3.2 for the feature extraction step of its computer vision workflow [36]. The standard OpenCV ORB keypoint detector and descriptor extractor is used [8]. A new ORB detector is instantiated for each image in the dataset. Producing ORB features from an image involves detecting keypoints and then further extracting feature descriptors from those keypoints.

As discussed in Section 2.3.2, the coins CADS works with often have irregularities around the edges that are irrelevant to determining die links. CADS ignores coin edges by using a user-defined radius outside which it will ignore all keypoints detected by the ORB detector mentioned above.[15] This radius is defined by the user in the CADS user interface. In between keypoint detection and descriptor extraction for each image in the dataset, keypoints outside this user-defined radius are ignored.

Figure 13a illustrates the necessity of filtering out edge features. As ORB feature detection relies on changes in pixel intensity to detect key points of interest [36], the edge of a coin produces several irrelevant keypoints, as seen by the black circles dotting the edge of the tetradrachm. Figure 13b shows the effect of keypoint filtering. The keypoints on the tetradrachm are constrained

[15] So long as all images in the dataset are centered properly, this method works well. There are plans to improve this process, possibly by detecting the edge of the coin with a circular Hough Transform, then filter out irrelevant keypoints in this manner [28]. Such a method will make CADS more robust to messier datasets, but is presently beyond the scope of this work.

to features of the type that a numismatist might also study: along the curve of Athena's ear, along her headdress, and along the outline of her neck.

(a) Without edge filtering (b) With edge filtering

Figure 13: Features extracted with and without edge filtering[16]

Once irrelevant keypoints have been filtered out, descriptors are created from the keypoints and feature extraction on the image is complete. CADS repeats this process for each image in the dataset, then automatically moves on to feature analysis.

4.4 Image Clustering

Feature analysis in CADS comes in the form of hierarchical clustering, which produces a tree structure containing potential die links. To better understand how CADS uses hierarchical clustering, it is necessary to discuss how features are matched, how the clustering algorithm determines the distance between two images based on their matched features, and how the algorithm produces the dendrogram.

4.4.1 Feature Matching and Distance Function

The CADS tool groups coins into potential die links by matching features. We first discuss how the coin features are matched, and what the strength of those matches conveys.

[16]The black circles were added manually by the authors in an image manipulation program for readability, as the small colored keypoints are otherwise difficult to see in print.

Algorithm 1 Custom distance function for hierarchical clustering

Input: Feature match pairs for two images

Output: Scalar value representing the distance between two images

1: $Matches = \langle (m_{1,1}, m_{1,2}), (m_{2,1}, m_{2,2}), \ldots (m_{n,1}, m_{n,2}) \rangle$

2: $Matches' = \text{sort}(Matches, (a, b) \Rightarrow \text{distance}(a) - \text{distance}(b))$

3: $BestMatches = \langle Matches'_1, Matches'_2, \ldots Matches'_{20} \rangle$

4: $\text{dist} \leftarrow \text{fold}(BestMatches, (a, b) \Rightarrow a + \left|\text{distance}(b)\right|) / \text{size}(BestMatches)$

As discussed in Section 3.3, feature matching is the process by which individual features extracted from two separate images are tested against one another for matches. CADS uses the Brute Force Hamming [5] method in OpenCV to match feature descriptors to one another. The name is suggestive of its methods — it tests each feature on one image to every other feature in the other image in a brute-force manner. The actual process of determining feature distance involves a comparison of the underlying feature vectors, which varies based on implementation.

Once this process is complete, CADS receives a list of feature matches which it can then feed into a custom distance function used by the hierarchical clustering algorithm to convert a list of feature matches into a single scalar value it can work with. A formal definition of this distance function is given below.

The distance function takes the output of the brute force feature matching, consisting of up to the twenty strongest feature matches between two coins, sorts them in descending order by match distance, then averages those distance scores into a single scalar value. The hierarchical clustering algorithm uses this custom distance function to determine how similar two coins are from one another as it is performing its grouping [31].

Figure 10 in Section 3.3 shows an example of two coins struck from the same die, but clearly with different levels of die wear, circulation, and even planchet shapes. Despite these differences, CADS is able to match several distinct features between these two images, with a distance score of 39.9 by way of the

Figure 14: Feature matches between coins of different dies

distance function. Note how most of the lines drawn between features in the images are parallel to one another, indicating that the features are relatively in the same layout from one image to the other.

Figure 14 illustrates what CADS encounters when two coins are struck from different dies. The two coins depicted are clearly from different dies, and the features matched between them are seemingly matched at random, showing no order or uniformity. The lack of similarity is corroborated by the distance function's output of 60.525 - much greater than that of the two coins struck from the same die.

From these examples it is clear that matching ORB features with the Brute Force Hamming method is successful in determining coins that were struck with the same die, making this feature matching approach an ideal foundation for hierarchical clustering.

4.4.2 Hierarchical Clustering

To produce the final result, CADS uses the Agglomerative Nesting (AGNES) algorithm, which starts with each individual coin in its own cluster, then at each iteration of the algorithm merges the two closest clusters into one, until all coins are in one large cluster [31]. In many ways hierarchical clustering captures

the nature of what numismatists do in their own manual die studies - they compare many coins to one another, sorting them by similarity to one another, until all coins are grouped with the ones they most closely resemble [7, 1]. The difference is that hierarchical clustering preserves the order (the *hierarchy*) in which coins were grouped together. Dendrograms are used in CADS to display the hierarchical clustering produced by the computer vision workflow, as they are a convenient and intuitive way to visualize hierarchical data.

Figure 15 illustrates what this output looks like in CADS. Numerical labels on intermediate tree nodes denote a dissimilarity score among all the node's children, giving a scalar measure of similarity among each coin below that node in the hierarchy. When comparing two intermediate nodes, a lower score on one node denotes that the coins contained below that node are more visually similar to one another than the coins in the other. It is for this reason that dissimilarity scores intuitively climb at each increasing height from the leaves, as a minimum of one additional coin is being considered in the dissimilarity calculation at each successive level.

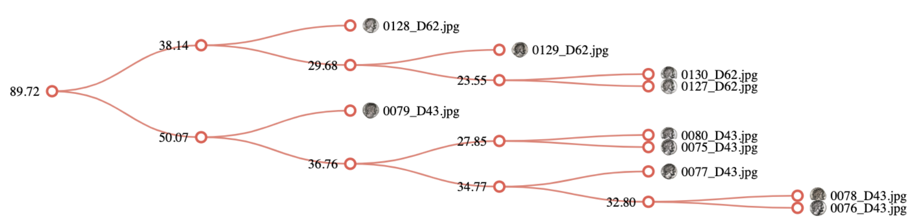

Figure 15: CADS output for coins from D62 and D42.

Figure 15 depicts a clustering of coins from two similar dies, and is thus fairly uniform with steadily increasing dissimilarity scores. Only the left-mode node, joining the two dies, exhibits a large jump. Figure 16 depicts a clustering of coins struck from multiple dies. The labels to the right of the coin images denote the die they belong to immediately following the underscore.

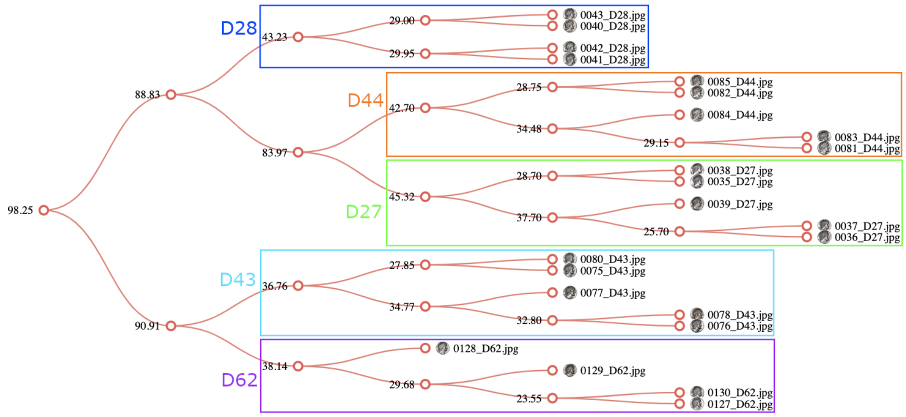

Figure 16: CADS output for coins from dies labeled D27, D28, D43, D44, and D62. Colored boxes are annotations added to the output.[17]

The colored boxes added to Figure 16 highlight the different die groups in the dendrogram. With this it is easy to follow along with hierarchical clustering intuitively. Note that at each successive level up from the leaves, coins from the same die have been linked together into a single node. No nodes containing coins from any one die were merged with nodes containing coins from another die before the entire die was merged into a single node. In this way, the agglomerative nesting algorithm has properly clustered the coins into their respective die groups as previously labeled by hand. It can be further noted that the dissimilarity scores jump significantly once nodes containing coins of a single die are merged with nodes containing coins from another die, indicating a sharp drop in coin similarity within the resulting cluster.

[17]The ability to add such annotations in CADS is planned in future work.

5 Results

This chapter will reflect on the results of our work on the Computer-Aided Die Study thus far. Recall that in Chapter 2 we divided a die study into the following 6 steps:

1. Gather data
2. Clean up images
3. Identify key comparators
4. Sort images into rough piles, refine
5. Identify and confirm die links
6. Further analysis (chronology, hands, etc.)

The CADS tool automates steps 3 and 4 in the above process, and provides a user interface for step 5. CADS automatically sorts coin images into an approximate clustering that the user can then use to effectively determine die links. This section presents metrics on CADS accuracy in determining die links, and details the user workflow that was developed for CADS.

5.1 Accuracy

We tested CADS against a known die study of 2$^{\text{nd}}$ century BCE Ptolemaic tetradrachms from Paphos, carried out by Dr. Thomas Faucher of the Centre National de la Recherche Scientifique and Dr. Julien Olivier of the Bibliothèque nationale de France [19]. Faucher graciously provided all 2,484 coin images and their associated die labels from the study to serve as a validation data set for CADS.

We tested CADS on a subset of 200 coins chosen sequentially from the die ordering used in the study. Table 1 below shows the results of running CADS

Adjustments	Correctly grouped by CADS	Accuracy (%)
None	181	90.5
Corroded & Damaged (CD)	184	92
Possibly Mislabeled (PM)	187	93.5
CD + PM	190	95

Table 1: CADS accuracy on 200 coins from Faucher & Olivier validataion set

on this subset. For the coins in the Paphos study, which have minimal wear and strong surface detail, CADS was able to determine die links with impressive accuracy. Coins were considered to be properly clustered by CADS if coins of the same die were extant in the sample, but they were first linked with coin(s) of a different die. Singletons, coins with no known die links in the study, were automatically considered properly clustered. We considered the possibility that one die could be split into two groups, but this did not occur on this sample. All errors were on the first link for a coin. This may be due to the coins from the Paphos study coming from a single hoard, with many coins from the same die having similar levels of circulation wear. This die splitting might occur frequently with data sets containing coins with more varying stages of circulation wear.

Using the above method for determining when CADS had clustered a coin correctly, a baseline accuracy of 90.5% was measured. However, certain coins in the data set were heavily corroded or damaged, with one in such condition having been given a die label of “X” by Faucher and Olivier in the original study. Such coins are difficult even for human eyes to determine a die label for, and are thus worth excluding from an accuracy measurement for CADS under normal conditions. When adjusted for corroded and/or damaged coins, an accuracy of 92% was measured.

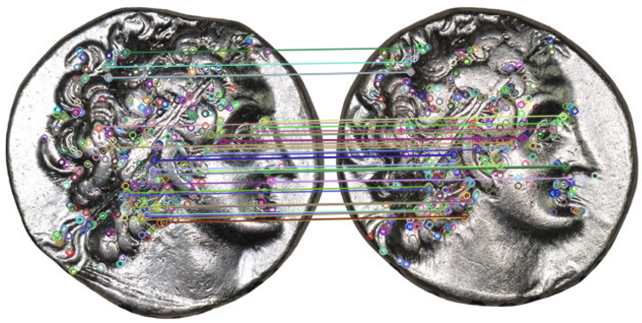

Figure 17: Die link found by CADS that disagrees with original study

There is yet another category of adjustment made to the accuracy measurement shown in Table 1: coins that are possibly mislabeled. Figure 17 shows one such example. In the original study, the coin on the left is labeled as belonging to obverse die 51, with the coin on the right labeled as obverse die 52. CADS clustered them both into obverse die 51 with a distance score of 31.1. This match was verbally confirmed by Dr. Faucher as equally plausible during a lecture on CADS delivered by the author on April 25, 2020 [43].

Coins mislabeled in the validation set, but properly clustered by CADS would of course be counted incorrect by the above methods, so an adjustment was made for coins that were deemed similar enough to possibly have been mislabeled. An examination of the results yielded several similar cases. With this adjustment, an accuracy of 93.5% was measured. With both the above adjustments applied to the baseline measurement, CADS was measured to have an accuracy of 95% when determining die links.

5.2 User Workflow

Through our work with CADS, we have produced a user interface with which users can effectively leverage computer vision technology to determine die links. This section will describe the current state of this process in CADS, to give a

feel for the workflow that has been developed.

The main steps of the user workflow in CADS are as follows:

1. Start a new study
2. Set user-defined parameters
3. Wait for results
4. Determine die links in output and annotate them in CADS

5.2.1 Start a New Study

Step 1 is initiated by the user by clicking the "New Study" option in the left-hand menu. The user is then prompted to browse for the folder containing the image files CADS will work with. The user selects the data set they wish to use for the study, then opens it with the file browser.

5.2.2 Set User-Defined Parameters

CADS then moves automatically into step 2, where the user is prompted to set the feature radius (Figure 18) where CADS will focus its feature extraction efforts. After the user confirms their choice, they are prompted by a similar window to define the median blur radius (Figure 19). Both windows contain detailed instructions, a live preview with a coin from their data set, and a visual example of what the user should aim to achieve when setting each radius.

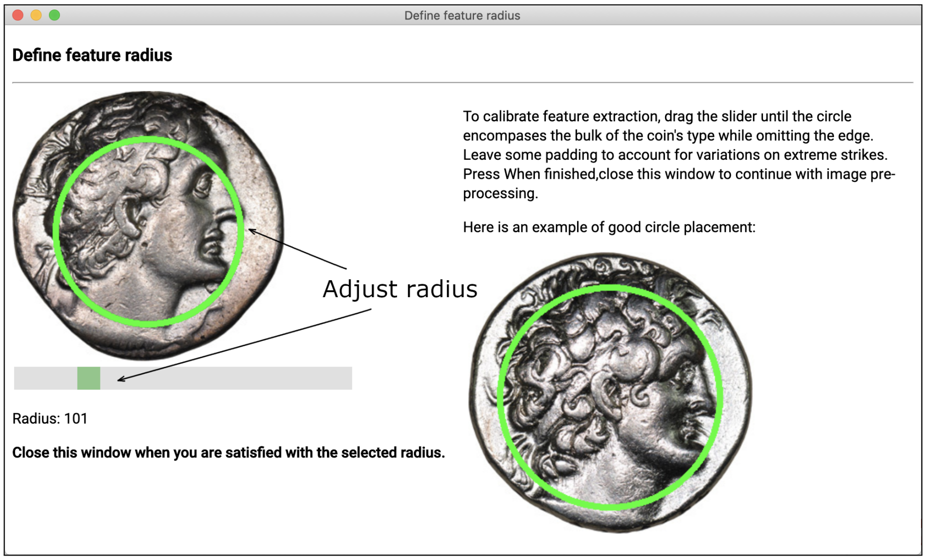

Figure 18: Adjusting the feature radius

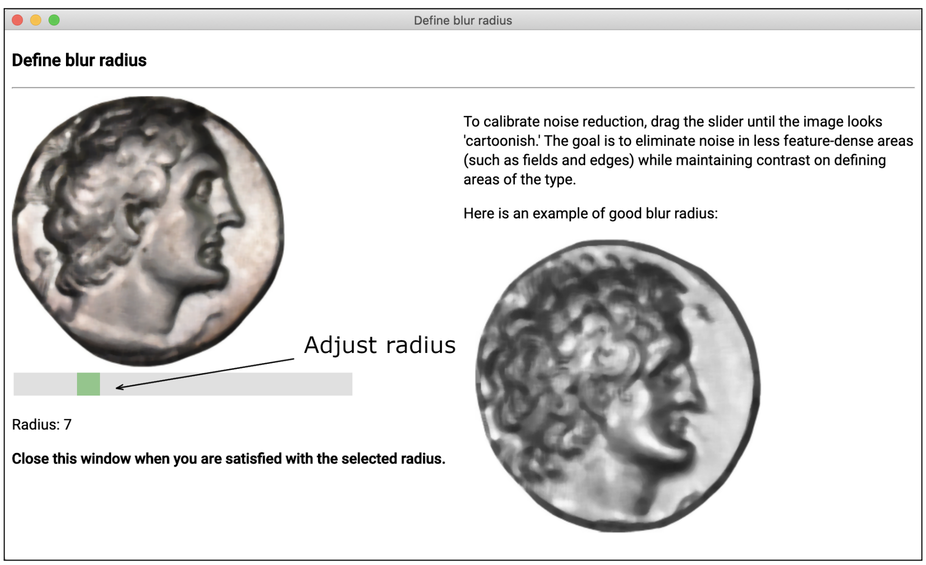

Figure 19: Adjusting the median blur radius

5.2.3 Wait for Results

Once a user completes step 2, CADS engages the computer vision workflow outlined in Chapter 4. This process can take anywhere from a few seconds to several hours, depending on the number of coins in the input and their quality.[18] Once CADS has finished clustering the coin images, it produces output in the form of an interactive dendrogram, depicted in Figure 20. The dendrogram is displayed in the center of the CADS program, as it is the main focus for the user. The user can zoom the dendrogram in and out using the scroll wheel of their mouse, and can drag the dendrogram around by holding down the left mouse button anywhere on the dendrogram display area.

5.2.4 Determine Die Links in Output and Annotate Them in CADS

Step 4 in the above workflow requires the highest level of user involvement. CADS was designed to assist the user in determining die links, but users must make such annotations themselves. CADS includes several user interface components which aid the user in this process. To the right of the dendrogram is the properties panel, which displays contextual information depending on which parts of the dendrogram have been selected.

[18]Higher resolution images cause all steps in the computer vision workflow to take longer, as there is more data to work with.

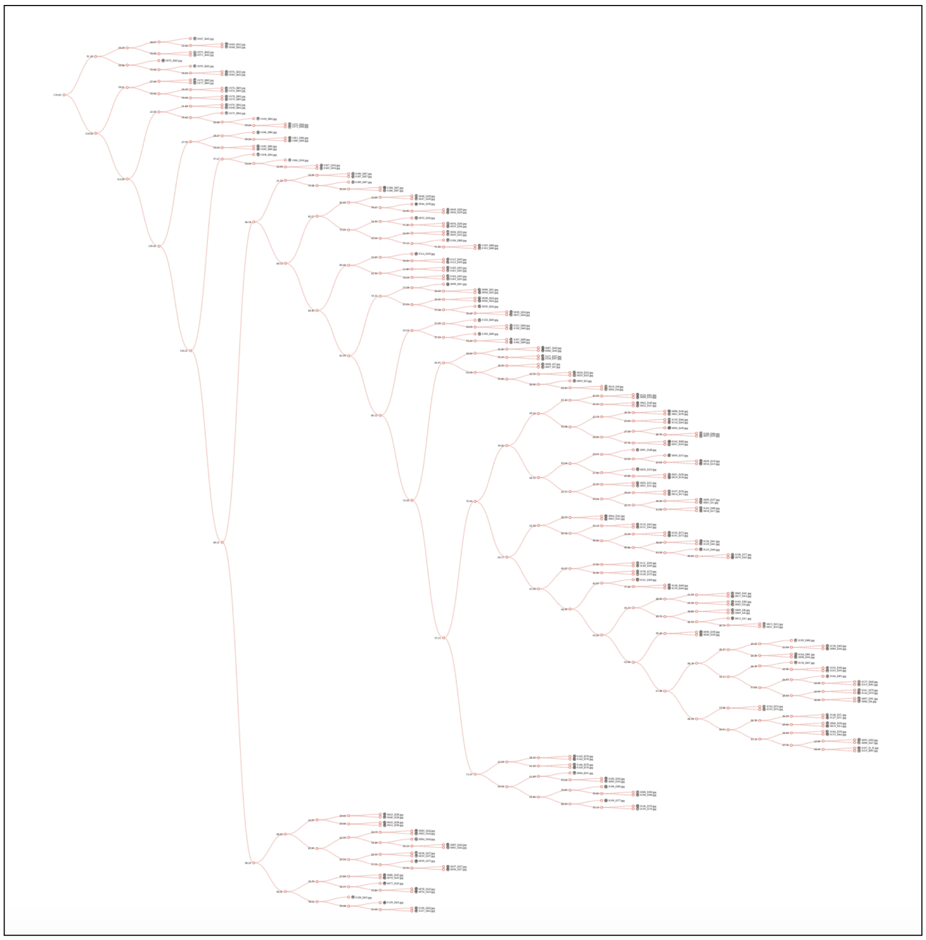

Figure 20: A zoomed-out view of CADS hierarchical clustering output

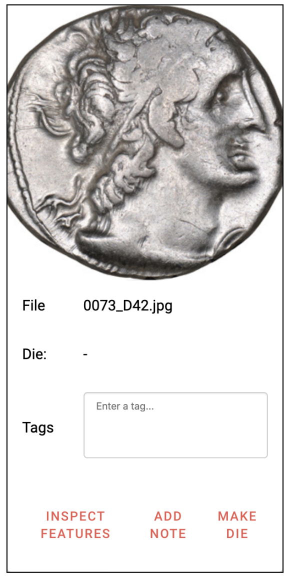

Figure 21: The properties panel when a leaf (coin) is selected

Figure 21 depicts the properties panel when the user clicks on a leaf node (representing an individual coin) of the dendrogram. It contains information related to the coin, a space for the user to add tags, as well as some action buttons on the bottom that allow the user to inspect the features extracted from the coin, add notes to the coin, or mark the coin as a singleton die.

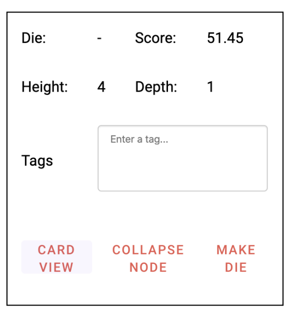

Figure 22: The properties panel when an intermediate node is selected

Figure 22 depicts the properties panel when the user clicks on an intermediate node in the dendrogram. As when a leaf node is selected, the properties

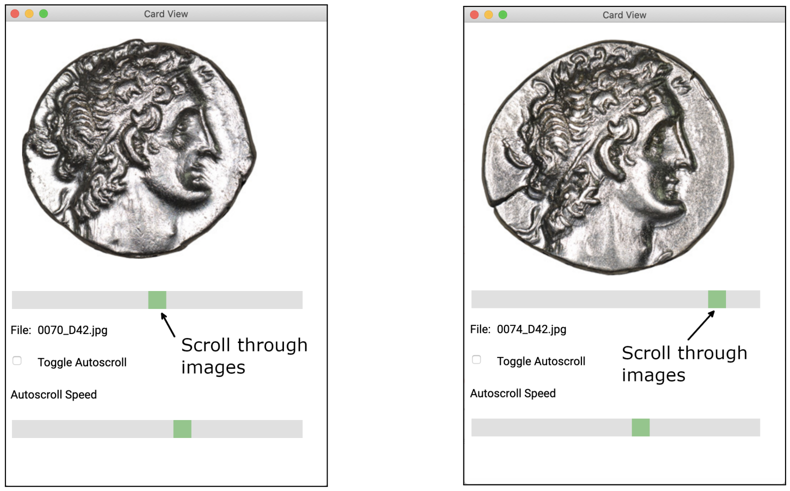

Figure 23: Using card view to inspect potential die groups

panel displays information about the node, a space for the user to add tags, and action buttons on the bottom that allow the user to collapse this node in the dendrogram, mark this node as a die, and open the card view.

The card view, depicted in Figure 23, is perhaps one of the most useful tools for determining die links in CADS. Reached from a selected intermediate node, the card view allows the user to view all the coins contained within the selected node. There is a scroll bar where the user can manually flip through images, as well as an option to enable auto scrolling and adjust the scrolling speed. The card view was designed with a traditional die study in mind - coin images pasted on note cards would be repeatedly flipped through to examine them for die links. The card view digitizes this process, making it easy to see die links where they exist in CADS output.

Once the user has determined a die link, it can be annotated using the "Make Die" action button in the properties panel, depicted on the left in Figure 24. CADS then applies the next available die label to the selected node and recursively to its children, until all coins in the selected node's sub-tree have been

labeled. The die label is then displayed in the properties panel, as depicted in the right image in Figure 24.

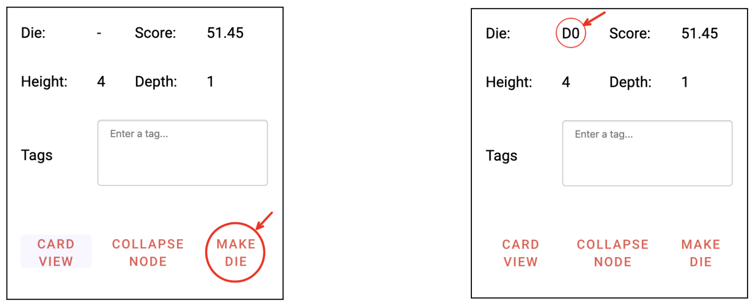

Figure 24: Marking a node as a die in the properties panel.

6 Conclusion

Our work with CADS demonstrates that a computational approach to die studies is effective. We achieved a 90.5%-95% clustering accuracy with a small validation set. The methods described in this thesis are a proof-of-concept, demonstrating that computer vision methods can be applied to the automation of numismatic die studies. In its current state, CADS is already accurate enough to have some use in numismatic die studies. Further refinement in both the computer vision process described in Section 4 and the user workflow described in Section 5 will lead to even greater accuracy in die link determination. This section discusses both the benefits of the CADS tool and future work.

6.1 Benefits of Computer-Aided Die Studies

Even with baseline accuracy of 90.5%, CADS can be considered state-of-the-art. The only alternative for a die study researcher is manual image comparison. As such, CADS could potentially eliminate a significant portion of the upfront work required when determining die links.

Table 2 illustrates how CADS improves on the traditional die study by automating the tedious upfront work that costs researchers valuable time. Recall that traditional studies, such as Faucher-Bossert's, can take upwards of a decade to complete [21, 47]. With CADS, an approximate clustering of several thousands coins can be performed in a few hours. The eyesight-destroying work has been automated, leaving the interesting parts — determining die links, chronology, etc. — to the researcher.

	By Hand	With CADS
Initial clustering	Months to years	Minutes to hours
Upfront work	Painstaking & strenuous	Automated
Collaboration	Physical files & co-location	File sharing
Constraints	Human lifespan & willpower	Computing power

Table 2: Comparison of traditional die study and CADS

Further, where traditional die studies are bound by human ability and lifespan, CADS is simply bound by computing power. Whereas a few thousand coins is a "monumental undertaking" [47] for a single researcher, CADS can handle the same volume of images overnight, vastly decreasing the time taken by the researcher to complete the study.

Beyond automation, CADS allows easier collaboration among die study researchers. CADS files can be easily shared among collaborators, with clustering, note, tags, and all other detail preserved. Traditional studies with physical files in careful organization were never so easily shared.

CADS opens numismatic die studies up to coin issues surpassing that of Fischer-Bossert's study, greatly widening the scope of what numismatists can learn from ancient coinages [21, 47]. CADS also reduces the time taken for die studies, and can even be used to validate die studies that have already been conducted. With further development, CADS has potential to become an indispensable component of the numismatic toolkit.

6.2 Future Work

While our work with CADS thus far has produced promising results, there is still much work to be completed before CADS will be ready for an initial release. Close collaboration with the numismatic community and feedback from

potential CADS users will be critical in determining next steps for development. This section presents the immediate path to an initial CADS release, as well as some possible directions for future work as gathered from members of the numismatic community.

Perhaps the most clear of next steps for CADS is preparation of its code base for an initial open-source release. CADS would surely benefit from an open-sourced approach to software development and design, and it would ensure that the tool would remain freely available for any and all possible research applications. Providing CADS as an open-source tool certainly poses risk of it being used for commercial purposes, but we expect the benefit to die study researchers to far outweigh the impact of commercial use.[19]

CADS would also benefit from a user study. According to Borkin *et al.*, "formal evaluations are a valuable measure to determine the effectiveness of visual representations and data encodings" [4]. Given that CADS is heavily focused on visual representation of hierarchical clustering data, it would be worthwhile to formally evaluate the effectiveness of its representation methods and improve them where possible.

We recently presented CADS to an audience of more than fifty leading numismatists through the American Numismatic Society's "Money Talks" lecture series [43], and received overwhelming support for the project. The questions raised during the lecture were helpful in identifying areas where CADS can be improved, as well as how numismatists intend to use it.

The following is a compilation of what was discussed for future CADS development:

- Make CADS more robust to different qualities of image input, such as those taken in different lighting conditions, at different angles, at different

[19] For an excellent discussion of the ethical considerations surrounding numismatics and the antiquities trade, see [3, 11, 15, 16, 17, 29, 44]

rotations, etc.

- Use a more sophisticated feature radius than a simple circle, possibly with the use of a Circular Hough Transform (CHT) [28].

- Add the ability to match sub-regions of the images, which would allow for analysis of features such as countermarks.[20]

- Add the ability to mask out certain regions of a coin the researcher would like CADS to ignore, like test cuts or countermarks.

- Test CADS on more data sets including incuse coins and other irregular issues that might identify where its computer vision workflow can be improved.

- Work with both obverse and reverse sides of the study at once, potentially aiding in establishing chronology and other more sophisticated analysis.

In addition to the improvements above, we also have plans to use CADS to perform a large die study in the near future, which will serve as the first test of the tool's effectiveness in a study where die links are not already known.

CADS is an exciting tool with potential to change how numismatists conduct die studies. There is still much work to be done before CADS is released, but one thing remains clear: as it continues to be developed, involvement from the numismatic community will be critical at every step.

[20] A *countermark* is a punch or stamp added to a coin after minting, often after it has entered circulation.

References

[1] American Numismatic Society. Introduction to numismatic terms and methods. http://numismatics.org/seminar/termsmethods/. Accessed: April 27, 2020.

[2] Anwar, H., Zambanini, S., and Kampel, M. Coarse-grained ancient coin classification using image-based reverse side motif recognition. *Machine Vision and Applications 26* (2015), 295–304.

[3] Beckmann, M. Numismatics and the antiquities trade. *The Celator 12*, 5 (1998), 34–38.

[4] Borkin, M. A., Gajos, K. Z., Peters, A. E., Mitsouras, D., Melchionna, S., Rybicki, F. J., Feldman, C. L., and Pfister, H. Evaluation of artery visualizations for heart disease diagnosis. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2479–2488.

[5] Bostanci, E. Is hamming distance only way for matching binary image feature descriptors? *Electronics Letters 50*, 11 (2014), 806–808.

[6] Bostock, M., Ogievetsky, V., and Heer, J. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2301–2309.

[7] Bracey, R. Introduction to die studies. https://www.youtube.com/watch?v=_y1THkRvU3I, 2013. Coins and Medals Summer School.

[8] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[9] Burrell, B., and Sheedy, K. A history of ephesus from coins: the greek state. http://humanities.mq.edu.au/acans/ephesus/chapters/chapter03_2.htm, 2002. Accessed: April 27, 2020.

[10] Carbone, L., and Yarrow, L. M. Opening access to roman republican die studies. *The Magazine of the American Numismatic Society 18*, 3 (2019), 7–19.

[11] Chippindale, C., and Gill, D. W. J. Material consequences of contemporary classical collecting. *American Journal of Archaeology 104*, 3 (2000), 463–511.

[12] Crawford, M. H. *Roman Republican Coinage*. Cambridge University Press, London, 1974.

[13] de Callataÿ, F. *Quantifying Monetary Supplies in Greco-Roman Times*. Edipuglia, Bari, 2011, ch. Quantifying Monetary Production in Greco-Roman Times: A General Frame, pp. 7–29).

[14] Electron. Electron platform. https://www.electronjs.org/.

[15] Elkins, N. T. A survey of the material and intellectual consequences of trading in undocumented ancient coins: A case study on the north american trade. *Frankfurter elektronische Rundschau zur Altertumskunde 7* (2008), 1–13.

[16] Elkins, N. T. The trade in fresh supplies of ancient coins: Scale, organization, and politics. In *All the King's Horses: Essays on the Impact of Looting and Illicit Antiquities Trade on Our Knowledge of the Past*, P. Lazrus and A. Barker, Eds. The Society for American Archaeology Press, Washington, D.C., 2012, pp. 91–107.

[17] Elkins, N. T. Ancient coins, find spots, and import restrictions: A critique of arguments made in the ancient coin collectors guild's "test case". *Journal of Field Archaeology 40*, 2 (2015), 236–243.

[18] Esty, W. W. *Quantifying Monetary Supplies in Greco-Roman Times.* Edipuglia, Bari, 2011, ch. The Geometric Model for Estimating the Number of Dies, pp. 43–58).

[19] Faucher, T., and Julien, O. *Egyptian Hoards I: The Ptolemies.* Institut Français d'Archéologie Orientale, Cairo, 2017, ch. Le Trésor de Paphos, pp. 258–478.

[20] Faucher, T., Téreygeol, F., Brousseau, L., and Arles, A. À la recherche des ateliers monétaires grecs: l'apport de l'expérimentation. *Revue Numismatique 6*, 165 (2009), 43–80.

[21] Fischer-Bossert, W. *Chronologie der Didrachmenprägung con Tarent.* Walter de Gruyter, Berlin, 1999.

[22] Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., and Crowley, M. Feature selection and feature extraction in pattern analysis: A literature review. *CoRR abs/1905.02845* (2019).

[23] Google. Material design guidelines. `https://material.io/design`.

[24] Howgego, C. *Ancient History from Coins.* Routledge, London, 1995, ch. Minting.

[25] Huang, T., Yang, G., and Tang, G. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing 27*, 1 (1979), 13–18.

[26] Huber-Mörk, R., Nölle, M., Rubik, M., Hödlmoser, M., Kampel, M., and Zambanini, S. Automatic coin classification and identification.

In *Advances in Object Recognition Systems*, I. Kypraios, Ed. IntechOpen, Rijeka, 2012, ch. 7.

[27] Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters 31*, 8 (2010), 651 – 666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

[28] Jain, N., and Jain, N. Coin recognition using circular hough transform. *International Journal of Electronics Communication and Computer Technology 2*, 3 (May 2012), 101–104.

[29] Kampmann, U. Who owns objects? a view from the coin trade. In *Who Owns Objects?: The Ethics and Politics of Collecting Cultural Artefacts*, E. Robinson, L. Treadwell, and C. Gosden, Eds. Oxbow Books, Oxford, England, 2006, pp. 61–76.

[30] Karypis, G., Han, E.-H., and Kumar, V. Chameleon: hierarchical clustering using dynamic modeling. *Computer 32*, 8 (Aug 1999), 68–75.

[31] Kaufman, L., and Rousseeuw, P. *Agglomerative Nesting (Program AGNES)*. John Wiley & Sons, Ltd, 1990, ch. 5, pp. 199–252.

[32] Kraay, Colin, M. *Archaic and classical Greek coins*. University of California Press, Berkeley, 1976.

[33] Marchal, T. Diagram of coin striking. `http://marchal.thibaut.free.fr/e_coinstriking.htm`, 2006. Accessed: December 12, 2018. Author's reproduction shown.

[34] McManus, B. F. Anvil die created by giovanni cavino - bibliothèque nationale. `http://www.vroma.org/~bmcmanus/coindies.html`. Accessed: April 27, 2020.

[35] Mehta, D., and Sagar, A. A survey on various techniques of coin detection and recognition. *International Journal of Computer Applications 69*, 5 (May 2013), 29–32.

[36] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision* (Nov 2011), pp. 2564–2571.

[37] Rui Xu, and Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks 16*, 3 (May 2005), 645–678.

[38] Saravanan, C. Color image to grayscale image conversion. In *2010 Second International Conference on Computer Engineering and Applications* (2010), vol. 2, pp. 196–199.

[39] Sasi, A., and K., S. Image based coin recognition system - a survey. *International Journal of Computer Applications 131*, 11 (December 2015), 19–22.

[40] Sathya, R., and Abraham, A. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence 2* (02 2013).

[41] Schaps, D. M. *The Invention of Coinage and the Monetization of Ancient Greece*. The University of Michigan Press, Ann Arbor, 2015, ch. The First Coins, pp. 93–96.

[42] Tareen, S. A. K., and Saleem, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (2018), pp. 1–10.

[43] Taylor, Z. M., and van Alfen, P. Coins and computation: New developments in the computer-aided die study. https://www.youtube.com/watch?v=wF6aZdhc0wg, 2020. ANS Money Talks: Numismatics Conversations.

[44] Tompa, P. K. Ancient coins as cultural property: A cause for concern? *International Journal of Legal Studies 4*, 1 (1998), 69–104.

[45] Unnikrishnan, G., and Sajith, S. P. Automatic coin recogniton using local spatial features. *IOSR Journal of VLSI and Signal Processing 3*, 5 (Dec 2013), 28–30.

[46] van Alfen, P. *Coinage of the Caravan Kingdoms.* American Numismatic Society, New York, 2010, ch. A Die Study of the 'Abiel' Coinages of Eastern Arabia, pp. 549–594.

[47] van Alfen, P. The computer aided die studies program. http://numismatics.org/pocketchange/cads/, 2017. Accessed: April 27, 2020.

[48] Velu, C. M., and Vivekanandan, P. Indian coin recognition system of image segmentation by heuristic approach and houch transform. *International Journal of Open Problems in Computer Science and Mathematics 2*, 2 (June 2009), 255–271.

[49] Vermeule, C. C. *Some notes on ancient dies and coining methods.* Spink & Sons, 1954. Reprinted from "The Numismatic Circular," 1953-1954.

[50] Wong, K. A short survey on data clustering algorithms. *CoRR abs/1511.09123* (2015).