Laurels and Crowns: Ancient Roman Coins Classification Using Computer Vision

Valérie BLANCH

2614867B@student.gla.ac.uk

University of Glasgow

September 2022

ABSTRACT

Due to the massive digitization of objects in museum collections, which has accelerated in recent years, new tools are needed to facilitate the process of filling databases and photographing artifacts [1].

This study proposes to apply computer vision to historical coins. It focuses on the coinage produced by the Roman Empire, and aims, through the use of convolutional neural networks, to distinguish the antoniniani from the denarii, which are two different denominations, by classifying their images.

Indeed, the antoninianus differs from the denarius, because on the former the emperors on the obverse wear radiate crowns, whereas on the denarius, are represented different sorts of hairstyles, for example laurel wreaths [8].

The purpose of this dissertation is therefore to classify these two different denominations by building six convolutional neural networks, capable of recognizing radiate crowns, thus facilitating the work of professionals in filling databases by automating parts of the digitization process.

Three of these convolutional neural networks were used as baseline methods, to be then customised to fit a dataset of 23,000 images, separated into three sets for training, validation, and testing. The first model was trained with grayscale images to test the impact of colour difference between classes, and despite significant results, appears to have slightly suffered from loss of information due to grayscale conversion.

A convolutional neural network composed of five layers and a pre-trained model based on InceptionV3 [47] gave the best results, and with the use of the grad-CAM algorithm [49], proved that they were indeed able to recognise the radiate crown or its absence during prediction.

However, the Grad-CAM heat maps also revealed a problematic bias due to the origin of the dataset: museums sometimes add their logo to the bottom of the image, or photograph the coins on various backgrounds, which brought concerns about the actual predictive abilities of the models.

This is why a second dataset was found and new predictions were computed from carefully curated images, without logos and on a consistent white background.

If InceptionV3 was the most efficient model for the first series of tests, it was the five-layers convolutional neural network that was the most efficient for the second test on a cleaner dataset.

After a more detailed inspection of the Grad-CAM visualisations, it seems that a relatively large number of errors produced by the models might be due to the bias of the initial dataset (logos and different backgrounds), and perhaps also the introduction of "divergent" coins (radiate crowns on denarii and their absence on antoniniani).

However, the results of the first and second tests remain particularly significant, with results averaging 98% for the test accuracies.

Furthermore, the insights brought by the Grad-CAM algorithm showed differences between denarii and antoniniani that had not been anticipated in the first place, adding new perspectives worthy of further investigation into domain knowledge.

TABLE OF CONTENTS

1. INTRODUCTION

The digitization of museum collections is a slow and laborious process aided by new technologies accelerating archiving. This work is all the more complex when it comes to cataloguing small and duplicable objects, for example, coins. For instance, the Smithsonian Museum has set up conveyor belts below cameras to facilitate digitizing its numismatic items [1].

The collation of data on historical coins involves not only the visual capture of the object but also the recording of its properties: an identification number, the date of its minting, its origin, its metal, its weight, its diameter, the description of the semantic elements it contains, and finally, its denomination. Projects like *nomisma.org*, which regroups hundreds of collections worldwide, give free access to such detailed databases [2].

For the Ancient era, the denomination of a coin is mainly determined by its material and size. Roman currency, for the imperial period (from 27 BCE to 476 CE for the Western Empire), can be composed of different materials: gold, silver, copper, and its multiple alloys [3].

The denomination system of ancient Roman coins is complex and has evolved over the centuries: through crises and economic upturns, new coins are introduced, and old ones are recalled and recycled; coins can also be debased or revalued upwards [4].

Scholars often study silver coins: being the primary currency struck to pay the armies [5], they have been regularly minted during the imperial era, found more often than gold coins, and in better condition than bronze or copper coins. Two denominations of silver coins are the most commonly found in hoards: the *denarius* and the *antoninianus* [6].

The denarius appeared during the Roman Republic, then gradually disappeared by the end of the third century CE, to be replaced by the antoninianus [4]. The reason for this monetary reform is a gradual debasement (decrease in silver content) of the denarius since the beginning of the Empire. It accelerated during the crisis of the third century. This crisis was political and economic, caused by epidemics, the ever-increasing military budget dedicated to protecting the borders, and, more generally, poor fiscal management [7].

When it started being minted at the beginning of the third century CE, the antoninianus was estimated at twice the value of the denarius: for the Romans to be able to make the distinction between the two denominations (since both coins were of similar material and size), the portraits of the emperors on the obverse of the antoninianus wear a "radiate crown"; on the denarius, the emperors mostly wear laurels, helmets, or are bareheaded [8].

Figure 1: From left to right: Augustus bareheaded on a denarius from 19 BCE, Trajan wearing laurels on a denarius of 103 CE, and an antoninianus of Caracalla wearing a radiate crown (215 CE).

Thus, it may be possible to distinguish the two denominations using image recognition. This dissertation aims to build a convolutional neural network capable of identifying radiate crowns from other forms of hairstyles on the obverse of ancient Roman coins. Thus, this model would ease the classification of denominations by automating the filling of museum databases for this variable in particular.

2. PREVIOUS RESEARCH

Numismatics is the study of physical currency in its many forms: coins, banknotes, various tokens, and medals. This field is fundamental to understanding our cultural heritage because money, and the symbols printed on it, provide invaluable information: for instance, Roman coinage was not only used for the exchange of goods and services, but it also relayed important events and reveals today how the emperors wanted to be perceived, as a primary vehicle for political ideologies [9].

In practice, computer vision applied to numismatics is mainly used by professionals – museums and auction houses – for classifying coins and detecting counterfeits [10].

The ever-growing portfolio of images of coins available on the internet makes the training of data-hungry models such as deep neural networks possible. Indeed, museums are not the only ones to digitize their collections: more than half a million coins are traded each year in North America, making professional dealers a consequent source of data as well [11].

Kim and Pavlovic [12] have built a convolutional neural network that classifies imperial Roman coinage by RIC type.

The *Roman Imperial Coinage* [13] is a catalogue grouping Imperial Roman coins by artistic similarity. This does not mean that all coins in one RIC are identical: the die that has struck them, the centring of the coin, and its state of conservation can be different, and those parameters make image recognition and classification more difficult for ancient coins than for modern machine-made currency.

Figure 2: Obverse ("heads") and reverse ("tails") of three coins belonging to the same type ric.1(2).aug.2.

Their model aims to identify the discriminative elements depicted on each face of the coins. To this end, they fine-tuned AlexNet [14] and compared their results with those of Kim and Pavlovic [15], a paper reporting on creating a Support Vector Machine model trained for the same task. The accuracy of AlexNet is 79.81%, while the SVM's accuracy remains at 60.66%.

Schlag and Arandjelovic [16] propose a model capable of recognizing the identity of the emperors, by taking advantage of the fact that the obverse almost always represents the face profile of the reigning emperor during this era. In their paper, they address the issue of the state of

conservation, which they prove has a significant impact on the model's accuracy. Furthermore, a bad centring of the die leads to a loss of information, and the emperors are not always represented by the same artists, nor always with the same features (facial hair or shaven, different haircuts, different clothes, different ages). The architecture of their model is inspired by the paper published by Simonyan and Zisserman [17], which is a model composed of 5 convolution blocks paired with max-pooling layers, followed by three fully connected layers.

Pan and Tougne [18] focus on the grading of the coins, an essential variable that, along with scarcity, determines the final price of the object. The grading considers a coin's conservation state and gives it a score. The problem with human grading is that it is subjective, time-consuming, and costly since it demands extensive expertise. The researchers use the AlexNet model as well. Their model was trained to detect damaged and smooth areas on the surface of the coins, with significant results.

Cooper and Arandjelovic [10] propose to update the paradigm underlying computer vision applied to coin recognition. Their paper criticizes the classification by type, which, according to the authors, has no practical application since the number of types is higher than the number of digitized ones; furthermore, it is still virtually possible to find new types not yet classified by the catalogues. Their approach differs from previous papers: their model recognizes the semantic elements separately rather than the coin type as a whole, meaning their model can recognise unknown coins.

They have built a convolutional neural network inspired by the architecture of AlexNet (five convolutional layers, three max-pooling layers and two dense layers), which recognizes small objects on the coins. They obtained a test accuracy of 0.84, 0.84, 0.72, 0.73, and 0.82 for the elements "cornucopia", "patera", "shield", "eagle" and "horse" respectively, which are the objects most often found on Roman coins.

3. EXPLORATORY ANALYSIS

The data used in this dissertation has been provided by nomisma.org [2]: it is a project created in 2010 by Andrew Meadows and Sebastian Heath, from the American Numismatic Society (ANS), which is an organization founded in 1858 and based in New York, with the purpose of educating and researching numismatics [19].

The initial objective of the project was to standardise numismatic concepts; since then, datasets from more than 50 collections from around the world have been donated to the project [20], under a Creative Commons Attribution 3.0 License [21], allowing use of the data for research purposes. Today, the database includes specimens from the collections of the ANS, the Münzkabinett in Berlin [22], the Ashmolean Museum in Oxford [23], the Bibliothèque Nationale de France in Paris [24], the British Museum [25] , and many more.

The data used for this dissertation has been downloaded through the SPARQL interface implemented on Nomisma's website [26]. As a result, 23,000 images of the obverse of Roman imperial coins have been downloaded, along with two tables, one dataset for the antoniniani and one for the denarii.

A. TEXT DATASET

The *antoninianus.csv* and the *denarius.csv* datasets contain the same variables but not the same number of observations: the *antoninianus* dataset is composed of 12,100 rows for 23 columns, whereas the *denarius* dataset contains 11,515 rows and 23 columns.

Variable	**Description**
Coin_Type	RIC type of the coin
Coin	ID of the object
Weight	Weight of the coin in grams
Diameter	Diameter of the coin in millimetres
Collection_URI	URI of the institution owning the coin
Collection	Name of the institution owning the coin
Obverse_URL	URL of the image of the obverse of the coin
Reverse_URL	URL of the image of the reverse of the coin
From_Date	Estimated date of minting – earliest bound
To_Date	Estimated date of minting – latest bound
Authority	Issuing Authority – usually the emperor
Deity	Deity usually depicted on the reverse
Denomination	Official value of the coin
Issuer	Artist who minted the coin
Manufacture	If the coin was struck or cast
Material	Precious metal contained in the coin
Mint	City where the coin was minted
Portrait	The person portrayed on the obverse
Region	Region where the coin was minted
Obverse_Legend	Legend written on the obverse
Obverse_Description	Short description of elements depicted on the obverse
Reverse_Legend	Legend written on the reverse
Reverse_Description	Short description of elements depicted on the reverse

Figure 3: Description of the variables contained in the text dataset.

The tabular datasets have been downloaded for two reasons: to download the image dataset using the *Obverse_URL* variable and to verify the representativity of the sample, in other words, whether the data matches the information found in historical literature.

First, the *Denomination* variable has been inspected, and missing values have been removed; all coins belonging to the antoninianus dataset are antoniniani, and coins in the denarius dataset are denarii.

The rows containing missing values for the relevant variables *Coin*, *Denomination*, and *Obverse_URL* have been deleted after downloading the text data. The variable *Metal* contains only the 'Silver' value and missing values. Those missing values could be deliberate and indicate that the coins have little to no silver in them while still belonging to one of the two denominations (which is the case for coins minted by the end of the time-period). Therefore, they have been kept in the datasets. The *Coin* variable has been tested for duplicates, and there is none.

Finally, obverse portraits that do not represent a male emperor have been omitted: indeed, sometimes a coin could represent a deity or an allegory on the obverse, particularly in the beginning of the empire, or the wife or daughter of the emperor, as a mean to ensure dynastic relevance [27]. Women on the obverse of the Roman imperial coins wear specific hairstyles like veils or diadems, but no wreath or radiate crown. Including women in the study would have been an interesting approach; however, they are greatly outnumbered by their male counterparts. Including them would have made the data unbalanced or too small to train an effective neural network.

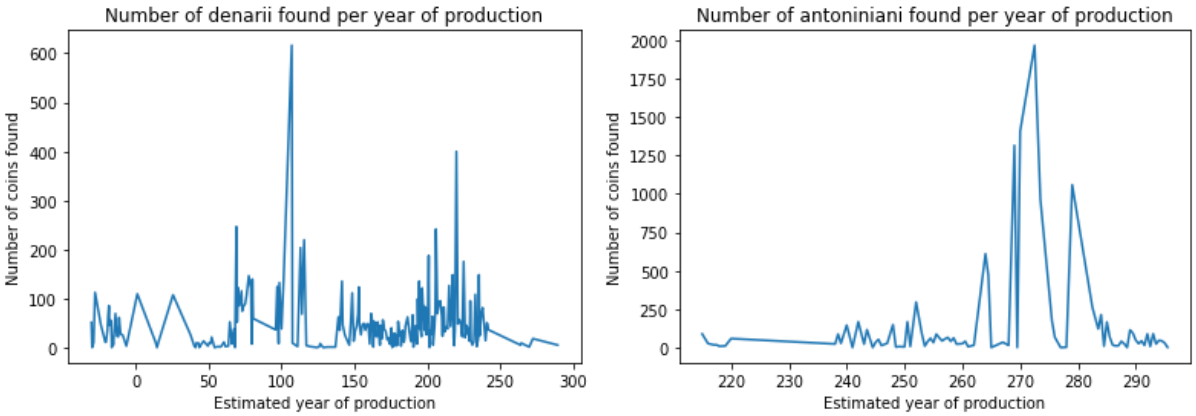


Figure 4: Time-series of the denarius and the antoninianus

The two time-series above have been drawn to verify if the sample downloaded from the nomisma.org website included the whole time-period, which seems to be the case. The denarius plot starts around the beginning of Augustus' reign and ends before the fourth century. The coins minted prior to the imperial period have been removed from the dataset.

The antoninianus slowly replaced the denarius as the principal silver coin throughout the third century. The x-axis ("Estimated year of production") has been computed by averaging the two bounds of the dating range ((From_Date+To_Date)/2). Interestingly, the denarius was minting for several centuries, although the antoninianus, in comparison, was relatively short-lived.

As we can see, the number of coins for both denominations is subject to significant variations over the years, which can be explained by many factors, the main one being the stochastic nature of the archaeological findings. Indeed, we can only infer from the number of coins found, but not the coinage produced [28]. Furthermore, the Roman empire witnessed multiple political crises during its existence, directly impacting its monetary regulations: emperors would increase the armies' wages to ensure their loyalty. Alternatively, new emperors would mint coinage first and foremost to disseminate their portrait and claim their power before being assassinated and replaced by a new candidate who would reproduce the same process [29]. For example, the year 238 CE, called the "year of the six emperors", well visible on the left plot, could explain an increased number of coins found for this year.

Although it has been said in the introduction that portraits on the antoninianus wore radiate crowns to avoid confusion with the denarius, there have been examples of radiate crowns on denarii. Using the text dataset, it was possible to find two radiate crowns from the denarius dataset.

Figure 5: Two denarius of type ric.1(2).cw.116 depicting Augustus wearing a radiate crown (68-69 CE).

The two coins, which belong to the same RIC type (ric.1(2).cw.116), represent Augustus and were struck around 68-69 CE. Augustus often represented himself as Apollo-Helios's protégé (the radiate crown being a "solar" symbol) [30]. At the time, the denarius was the only silver coin, so it is not surprising to see a radiate crown in that context.

The two coins have been kept in the dataset to see how the model would react to them.

On the other hand, several antoniniani were not marked as "radiate" in the dataset, but after further investigation, it seems that the text entry was erroneous. Therefore, we can assume there is no hairstyle different from radiate crowns in the antoninianus dataset.

Bar plots have been drawn to account for the hairstyles depicted on the two denominations.

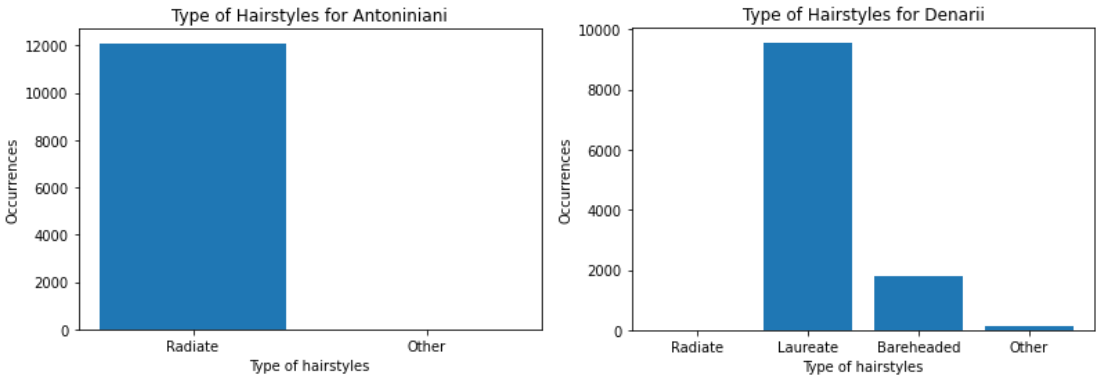


Figure 6: Occurrences of hairstyles per denomination.

As we have seen, there are only radiate crowns in the antoninianus dataset. However, the denarius dataset is way more varied regarding hairstyles. The laurel wreath is by far the majority. The "Other" category regroups a multitude of hairstyles, different types of diadems and wreaths (oak, for example), and helmets. Then it still contains two coins depicting a radiate crown.

Finally, summary statistics of the diameters and weights of the antoniniani and denarii have been calculated for comparison. The following table, computed from the text dataset, proves the necessity for the mints to differentiate the antoninianus from the denarius, as their measurements were very similar.

Diameter (mm)	Denarius	Antoninianus	Weight (g)	Denarius	Antoninianus
Count	9808	10742	Count	11501	12036
Mean	18.03	20.34	Mean	3.18	3.24
Std deviation	3.97	2.31	Std deviation	0.79	0.95
Minimum	0.00	0.00	Minimum	0.00	0.00
1st quartile	18.00	18.77	1st quartile	2.93	2.59
Median	**18.50**	**21.00**	**Median**	**3.18**	**3.24**
3rd quartile	19.00	22.00	3rd quartile	3.39	3.83
Maximum	200.00	81.50	Maximum	54.96	32.96

Figure 7: Summary statistics for diameter and weight per denomination.

The median of the diameter of the antoninianus is 2.5 mm greater than the median of the diameter of the denarius. Similarly, the weight of the denarius is 0.06 g lighter on average than for the antoninianus. The counts of the coins measured in the table above are not equal to the total number of rows in the datasets due to missing values. Furthermore, the minima (all equal to 0) must be understood as missing values, whereas the maxima are most probably mistakes made during the data entry. The corresponding coins have been verified and did not present any particularity that would have justified removing them from the image dataset.

B. IMAGE DATASET

The 23,000 images from the dataset have been collected using the *Obverse_URL* variable. This number does not correspond to the total number of rows in the tabular datasets due to 404 errors. Obtaining equal and round numbers for each denomination was also necessary to ease the batching process during training. The coins used in the present dissertation come from the same dataset and are subject to the same CC license, allowing use for academic research.

Below are presented several examples of the coins contained in the dataset. The first two rows are antoniniani, while the last two are denarii. As we can see, some of them are tagged by the collection owning them, which might be a potential issue, as it seems the tags are more often present in the pictures of the antoniniani. This aspect has been studied in more detail in the following chapters of this dissertation.

Figure 8: Examples of antoniniani (first two rows) and denarii (last two rows).

Another discovery has been made while exploring the image dataset: the antoniniani seem significantly less “silvery” than the denarii. Indeed, while the antoninianus was meant to be a remedy for the debasement of the denarius, it only contributed to the progressive decline of the silver currency [4].

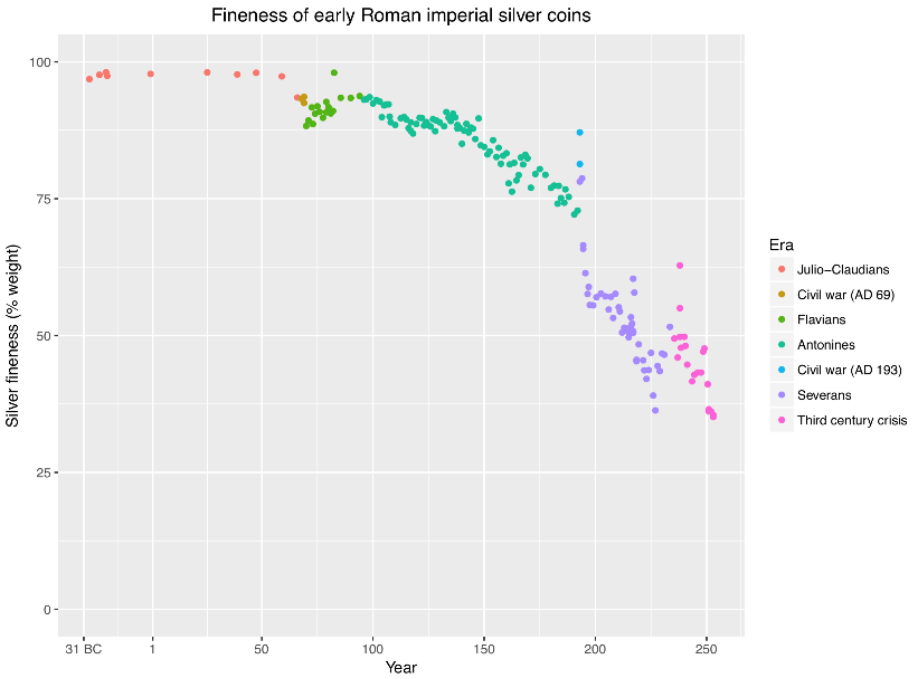


Figure 9: Fineness of Roman Imperial silver coins by Nicolas Perrault III - Wikipedia [31].

To inspect the colour difference between the coins, averages of the pixels of the images have been computed for each denomination, then converted back into images.

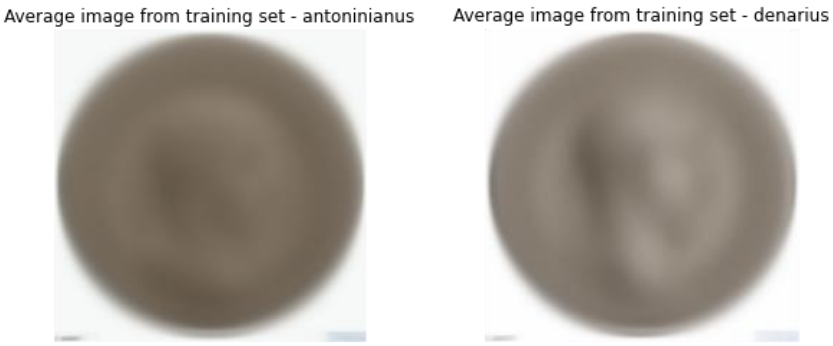


Figure 10: Average image of the antoniniani and the denarii in the training set.

The antoniniani seem darker and more saturated in colour due to the base metal being mainly bronze, while the denarii are greyer since their silver content was higher in the beginning of the Empire [4]. The colour variation adds an interesting parameter to the study: this dissertation assumes that the hairstyles are the discriminative features to be recognised by the models. However, the colour of the coins, as well as other elements, might affect the classification. Therefore, using an algorithm such as Grad-CAM seems necessary to analyse the models' choices and the features that actually trigger them.

The background and tags seem slightly darker for the "averaged antoninianus", which could be a potential issue for the ability of the model to generalise and make predictions on a different dataset. For instance, the coins in the collection of the University of Graz [32] are pictured on a green background.

For now, we will assume that the tags and backgrounds are uniformly distributed in both classes since both denominations come from the same collections and roughly in the same proportions.

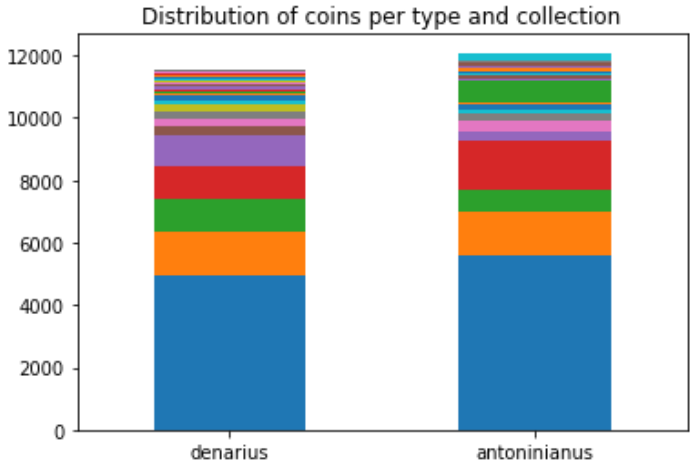


Figure 11: Distribution of denominations per collection.

4. PROCESS

To be able to classify images of coins into the two classes “antoninianus” and “denarius”, different architectures of neural networks have been built. In total, six models have been trained and tested. Their metrics (loss and accuracy) have been compared to choose the most efficient model.

Artificial Neural Networks are deep learning models assembled in a sequence of layers stacked together, the first being the input layer and the last being the output layer. The layers between the input and output layers are called “hidden layers”: each processes the information that passes through them according to parameters that are successively updated at each training epoch [33].

Convolutional Neural Networks are neural networks that are mainly used in the computer vision field. They contain convolutional layers, which, unlike fully connected layers, detect local features; stacked successively, they can progressively learn more and more complex patterns, using the inputs from the previous convolutional layers [33].

TensorFlow [34] is a machine learning platform, available for Python and R users, free and open source since its deployment by Google in 2015. It is the most widely used library for neural networks, for it is versatile, can run on any hardware, and automatically takes care of some underlying mathematical operations, like gradients [33].

Keras [35], on the other hand, is an API built on top of TensorFlow, created in 2015 by François Chollet. Its popularity is due to its user-friendly syntax for building and training neural networks. For instance, metrics, loss functions, and optimizers are already implemented and do not need to be defined from scratch [33].

The models presented in this dissertation have been built and trained using TensorFlow and Keras and one of the GPUs provided by Google Colaboratory. This Google project gives direct access to Python-based notebooks from a web browser [36].

A. CNNs WITH GRAYSCALE IMAGES

The first network built for this study is a model classifying grayscale images to inspect the impact of the colour of the coins on classification. Converting the images to grayscale seems to make the dataset more uniform, as shown in the figure below, which are the same coins presented on page 7 but converted to grayscale. The purpose of this manipulation is for the neural network to focus on the semantic elements of the coins instead of the colours. However, the antoniniani still seem darker than the denarii.

Figure 12: Antoniniani and denarii converted to grayscale.

A.1. MODEL 1

The image dataset has been divided into three sets, a training set containing 14,000 images, a validation set containing 7,000 images, and a test set containing 2,000 images. Mini-batches have also been defined for faster training [37]: 140 batches of 100 images for the training set and 70 batches of 100 images for the validation set. This setup is the same for all models presented in this project.

Data augmentation has been implemented to compensate for the relatively small size of the dataset: random rotation up to 40° and mirroring ("horizontal flip" in Keras). The other reasons for those modifications are for the model to recognise coins even when professionals picture them from different angles. Furthermore, we have seen that the dies are not always correctly centred when the coins are struck. Finally, although the portraits on the obverse of the coins are most often looking to the right, several examples in the dataset show portraits depicted looking to the left.

The images have been converted to grayscale and measure 150x150 pixels. The pixel values (initially ranging from 0 to 255) have been normalised to fit a range between 0 and 1, for the model to converge faster by feeding it with smaller values [38].

Therefore, the first convolutional neural network to be built, trained, and tested as a baseline model is a simple architecture given by Chollet as an example to classify the MNIST digits dataset [33].

MODEL 1			
LAYER	KERNEL SIZE	N° OF KERNELS	ACTIVATION
Convolutional	3 x 3	32	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	64	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	128	ReLU
Flatten			
	OUTPUTS		
Dense	2		Softmax

Figure 13: Architecture of Model 1 (Grayscale Baseline Model) by Chollet F.

A convolutional layer convolves the pixel values of the input with the pixel values of a filter, called a kernel, to detect edges. Thus, they are enhanced in the output of that particular layer [38]. Here, the first convolutional layer contains 32 kernels that measure 3x3 pixels each.

The amount of filtering, that is, the pixel values of the filters, can be either handpicked or initialised randomly, then calculated during forward propagation using a non-linear activation function. The ReLU function has been used; ReLU stands for *rectified linear unit* and is a commonly used function whose gradient is 1 for all positive values [38]. The formula of the ReLU function is [39]:

$$R(x) = \max(0, x)$$

The max-pooling layers reduce the output size, decreasing the number of parameters that need to be trained, simplifying the model, and accelerating the training. The input is divided by windows (also called kernels), and the maximum pixel value in each kernel is selected. That way, the essential information is kept. For this architecture, the kernel size is 2x2 pixels for each max-pooling layer [38].

Since the output of the last convolutional layer is a 34x34x128 matrix, it needs to be flattened before being fed into the network's last layer because it is fully connected. The output layer is a dense layer containing two units (for a binary classification: either "antoninianus" or "denarius"). The Softmax activation function, which is a generalisation of the Sigmoid function, normalises an input vector into a probability distribution summing up to 1 [38]. The formula of the Softmax for a binary output can be rewritten as a Sigmoid function [39]:

$$S(x) = \frac{1}{1 + e^{-x}}$$

Instead of using a Sigmoid activation function for the output layer, which is the function traditionally used for binary classification, a Softmax function has been implemented instead: the package used in this dissertation to compute Grad-CAM visualisation*, tf-explain* [40], only seems to work with a Softmax implementation [41].

During each epoch, the network engages in forward and back propagation: the training process is a loop that iteratively calculates weights with the activation function (forward pass) and then adjusts them (backward pass) to minimise the loss and increase the accuracy of the model [33].

Cross-entropy has been used for this model: the loss function measures the errors between the proper labels and the predicted ones, then produces a probability value that increases or decreases

proportionally to the model's performance. Ideally, the loss should be as small as possible. The formula for one example, with target y and prediction yhat is [39]:

$$L = \log\left(1 + \exp(-y \times \hat{y})\right)$$

The optimizer chosen by Chollet is RMSprop: optimizers update the networks' weights according to the results produced by the loss function. RMSprop, in particular, adjusts its learning rate based on previous calculations to speed up the training process [42].

The model has been trained for 50 epochs, with a callback option added to save the best model with the lowest validation loss. The loss has been used as a determining metric for choosing the "best model": while the accuracy reports on the model's ability to predict, the loss metric indicates its confidence in doing so. The same process has been implemented for all models presented in this project.

The training (388,610 parameters) lasted 46 minutes and 9 seconds (wall time). Although the baseline model gave significant results, with a validation loss of 0.1015 and a validation accuracy of 0.9741, the model has been modified to lower the validation loss: it seems relatively high compared to the training loss. The model might not be adequately calibrated, because at the same time the validation accuracy stays close to the training accuracy. Miscalibration is a discrepancy between a model's confidence and accuracy, potentially leading to poor predictions on new data [43].

This phenomenon can be seen in the two graphs below, representing the validation and loss for both the training and validation sets computed at each epoch.

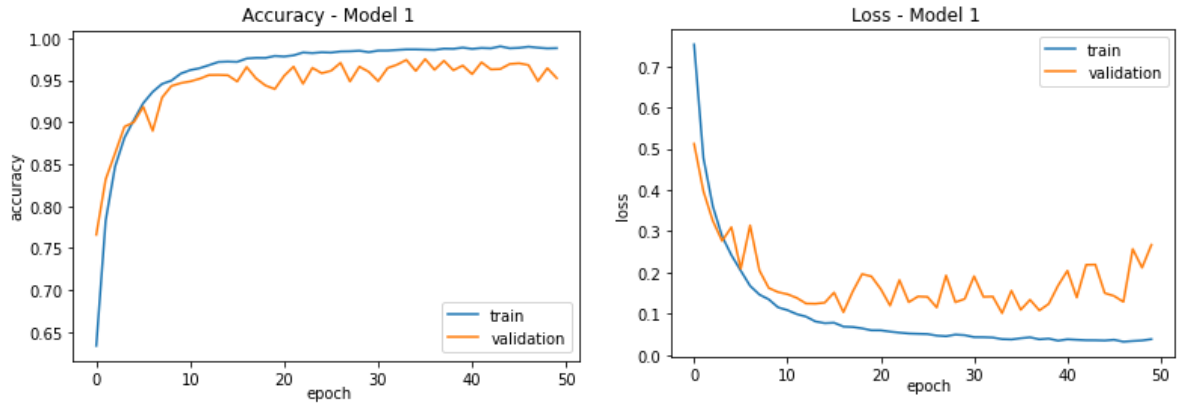


Figure 14: Accuracy of Model 1 (left) and Loss (right) for the training and validation set.

A.2. MODEL 2

The second model built for this study is a modified version of Chollet's model described above.

MODEL 2			
LAYER	KERNEL SIZE	N° OF KERNELS	ACTIVATION
Convolutional	3 x 3	32	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	64	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	128	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	256	ReLU
Max-pooling	2 x 2		
Flatten			
Dropout 0.5			
	OUTPUTS		
Dense	2		Softmax

Figure 15: Updated CNN (Model 2) w/ grayscale images as input.

Like the first model, the input layer processes 150x150 pixels grayscale images, with the same normalisation of the pixel values and the same data augmentation (rotation and horizontal flip). A max-pooling layer with a kernel of size 2x2 has been added to the third convolutional layer, then another pair of convolution layer with 256 kernels of size 3 x 3 and max-pooling layer with a 2x2 kernel.

To compensate for the relative depth of the model, a dropout layer has been added after the dense layer to avoid overfitting: it removes units randomly to prevent the model from getting distracted by noise and becoming overspecialised, hence being better at generalising. A dropout of 0.5 means that 50% of the units in the flatten layer will be dropped out randomly [38].

ReLU has been used as an activation function for all concerned layers and Cross-entropy for the loss function. The Adam optimizer has replaced RMSProp: it is similar to RMSProp as it adjusts learning rates during training, except its computational cost is lower [44].

After 50 epochs (for the training of 412,930 parameters), the second model's "best" weights have been saved: its validation accuracy is 0.9887, and its validation loss is 0.0392, making it potentially more efficient than the first model. The graphs of the accuracy and loss for both the training and validation sets show relative stability, no obvious overfitting, and better calibration than for model 1. The training lasted for 45 minutes and 47 seconds (wall time).

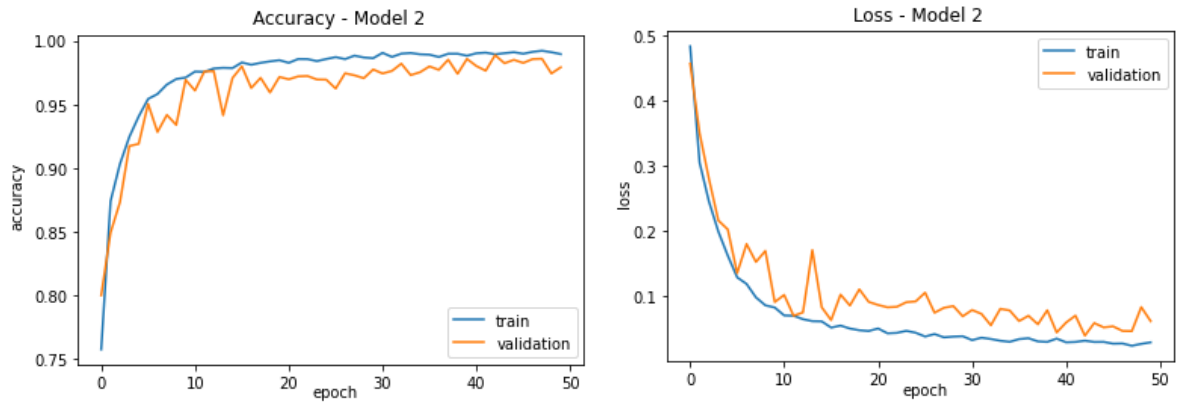


Figure 16: Accuracy and Loss for the train and validation sets (Model 2).

B. CNNs WITH RGB IMAGES

The first two models have been fed with grayscale images due to uncertainty regarding the various colours of the coins. Another advantage of using grayscale is the relatively low computational cost: using Colab's GPU, none of the training sessions exceeded one hour.

Even though the first two models have shown promising results during the training process, new models using RGB images as input have been built since the grayscale conversion might have resulted in a loss of information.

At this point, building more models aims to maximise accuracy and inspect the colours' impact on the classification. In other words, we want to assess if the following models will discriminate coins based on the hairstyles of the emperors regardless of the metal, which is the premise underlying this dissertation that will be tested later with grad-CAM.

B.1. MODEL 3

The third model used is another CNN presented by Chollet [33], used as an example for the "Dogs vs. Cats" challenge that took place on the website Kaggle in 2014 [45]. It will be used as a baseline for the RGB-based models built in this study.

Apart from the grayscale conversion, the same pre-processing of the images has been implemented: normalising the pixel values, data augmentation with mirroring, and 40° rotation.

The only difference between the architecture of Model 2 and 3 is the removal of the dropout layer after the flatten layer and the addition of a convolutional layer containing 256 kernels of size 3. The optimizer used by Chollet is RMSProp and the loss function cross-entropy.

The number of parameters is higher (991,298) since the input shape is a 150 x 150 x 3 matrix this time. As a result, the training lasted longer than the two previous models (1 hour, 16 minutes, and 53 seconds wall time).

MODEL 3			
LAYER	KERNEL SIZE	N° OF KERNELS	ACTIVATION
Convolutional	3 x 3	32	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	64	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	128	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	256	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	256	ReLU
Flatten			
	OUTPUTS		
Dense	2		Softmax

Figure 17: Architecture of baseline Model 3 (Chollet)

The optimal weights were saved when the validation loss hit 0.0573 for a validation accuracy of 0.9823. Comparably to the first model, there is a mismatch between the validation accuracy, which closely follows the training curve, and the validation loss, more unstable, hinting at a potential calibration issue. Therefore, a fourth model has been built based on Chollet's example.

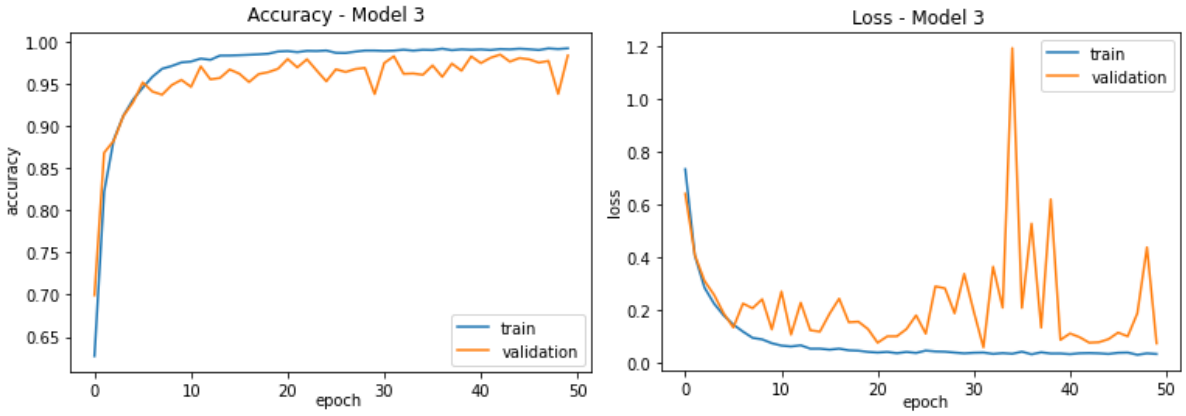


Figure 18: Accuracy and Loss per epoch for the training and validation set (Model 3).

B.2. MODEL 4

The optimizer used for Model 4 is Adam, and cross-entropy for the loss function. As for the other models, the convnet has been trained for 50 epochs.

MODEL 4			
LAYER	KERNEL SIZE	N° OF KERNELS	ACTIVATION
Convolutional	3 x 3	32	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	64	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	128	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	256	ReLU
Max-pooling	2 x 2		
Convolutional	3 x 3	256	ReLU
Max-pooling	2 x 2		
Flatten			
Dropout 0.5			
	OUTPUTS		
Dense	2		Softmax

Figure 19: Architecture of Model 4.

It is similar to the third model, except a max-pooling layer has been added after the last convolutional layer and a dropout of 0.5 after the flatten layer, in an attempt to stabilise the loss. The training time was similar to model 3 (1 hour 15 minutes and 6 seconds wall time).

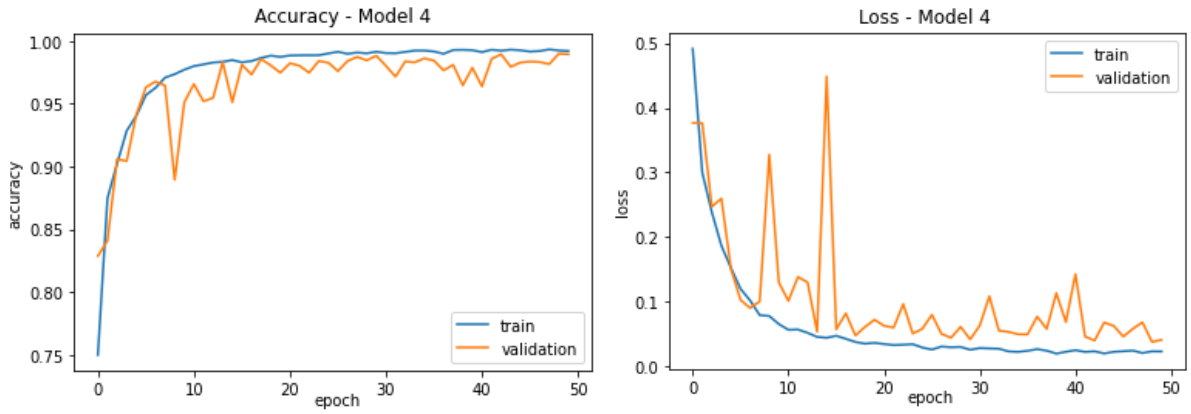


Figure 20: Accuracy and Loss per epoch for the training and validation sets (Model 4)

The weights saved by the callback implementation gave a model with a validation accuracy of 0.9897 and a validation loss of 0.0379. The added layers (max-pooling and dropout) seemed to have lowered the validation loss indeed. We can then assume that model 4 is an improvement from the baseline model 3.

C. PRE-TRAINED CNNs WITH INCEPTIONV3

Since most of the literature presented in this study was based on fine-tuning AlexNet, it has been decided to try implementing transfer learning for this study using the architecture and trained weights of InceptionV3.

The InceptionV1 model was published in 2014 by Szegedy et al. [46] to bring a new paradigm to the deep learning community by solving issues commonly faced while building very deep convolutional neural networks. Those problems are usually overfitting, expensive computational cost, and the difficulty in choosing the right kernel size when the objects we want to detect in an image have different locations or sizes.

The authors have designed an "inception module": it contains multiple convolutions with different kernel sizes, including 1x1 to reduce dimension, before a max-pooling layer, to lower computational cost. The output is concatenated before being passed on to the next module. Therefore, the model gets progressively "wider" instead of "deeper". InceptionV3 was presented in 2015 [47] along with InceptionV2, bringing several upgrades, such as kernel size, allowing for a smoother decrease in dimensions.

C.1. MODEL 5

The baseline method used for transfer learning with InceptionV3 is an example given by Moroney L. [38]. InceptionV3 is part of the Keras applications library [48], which makes it easy to load and customise. The model has been loaded up to the seventh module by Moroney. The layers have then been frozen, preventing the weights from being retrained.

Several layers have been added to the pre-trained base: a flatten layer to fit the output into a fully connected layer of 1024 units with a ReLU activation, then a dropout layer of 20%, and finally the output layer of 2 units with a Softmax activation.

This model was implemented with a cross-entropy loss function and an RMSProp optimizer with a learning rate of 0.0001. The model has been trained for 50 epochs for a wall time of 1 hour, 16 minutes, and 11 seconds for 38,538,242 trainable parameters.

The callback function has saved the weights with the lowest validation loss, which is 0.0731, for a validation accuracy of 0.9766. The graphs of the loss and accuracy per epoch provide interesting insights:

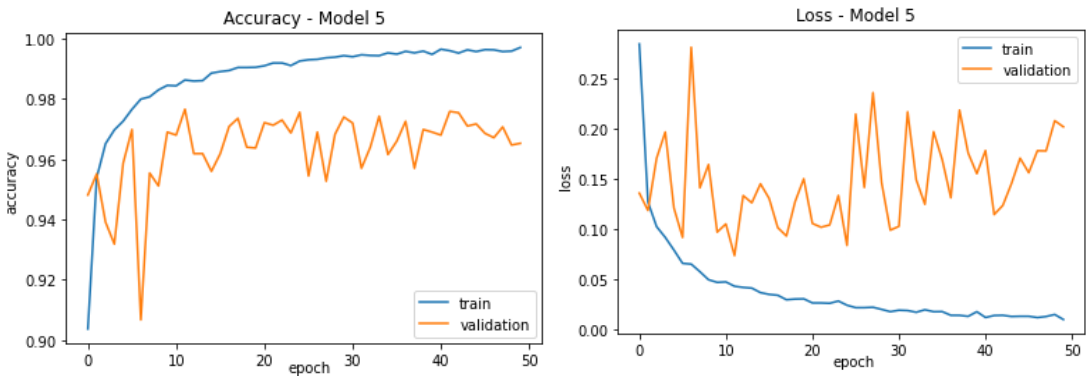


Figure 21: Accuracy and Loss per epoch for the training and validation set (Model 5).

If we compare these plots with the previous ones, the fifth model's validation accuracy starts way higher at the end of the first epoch (0.9481). The validation loss starts way lower than for the other models as well (0.1355): this is because the weights of the modules have already been trained, which is the main advantage of pre-trained models. However, the model seems to overfit immediately, which calls for fine-tuning. This is why a sixth model has been built for this dissertation.

C.2. MODEL 6

The last model presented here is a modification of Moroney's fine-tuned model. To avoid overfitting, instead of building a model with InceptionV3 up to the seventh module, the model only uses it up to the fourth module. This choice has been made after progressively removing modules until an optimum (lowest validation loss without overfitting) has been reached. Then a convolutional layer with 128 kernels of size 3 x 3 and a ReLU activation function has been added, followed by a max-pooling layer with a kernel of size 2 x 2, a flatten layer, a 0.5 dropout layer, and finally, an output layer of 2 units with a Softmax activation. The Adam optimizer was used (with the default learning rate of 0.001), along with the cross-entropy loss function.

The training lasted 1 hour, 15 minutes, and 52 seconds for 50 epochs for 885,890 trainable parameters. The model does not seem to overfit anymore.

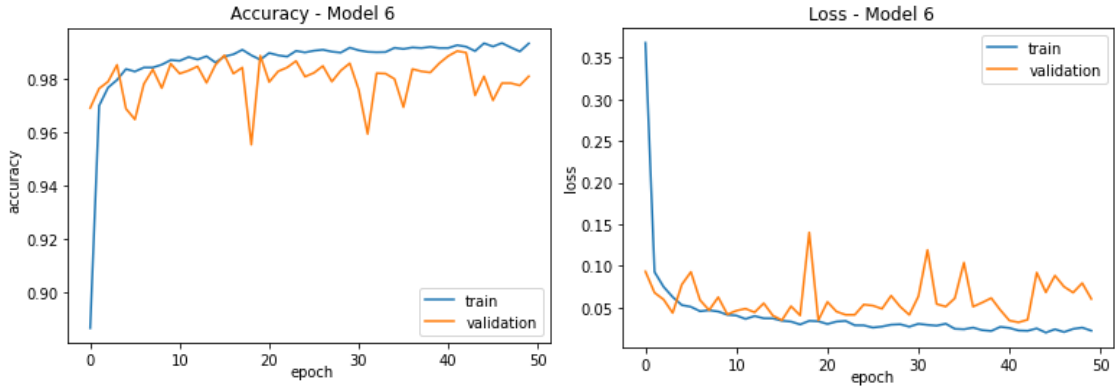


Figure 22: Accuracy and loss per epoch for the training and validation sets (Model 6).

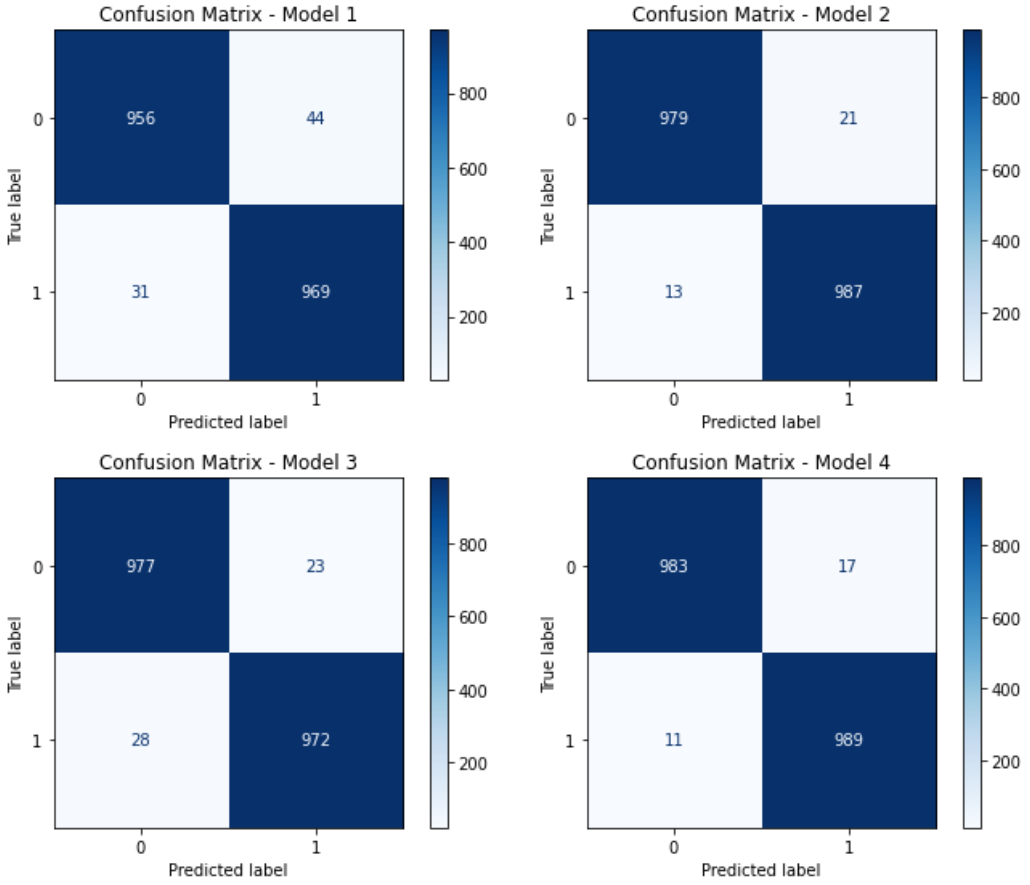
Once again, transfer learning is visible on the accuracy and loss graphs: the validation loss is at 0.0932 at the end of the first epoch for an accuracy of 0.9690. The most performant weights saved during training give a model with a validation loss of 0.0322 and a validation accuracy of 0.9903, the highest accuracy of all the models presented in this dissertation.

5. RESULTS

Out of the 23,000 coins in the dataset, 2,000 (1,000 denarii and 1,000 antoniniani) have been kept aside to test the predictive performances of the six models. To get more insights into the decisions made by the model, Grad-CAM visualisations have been computed. Because of the ambiguous information gathered from this process, a new dataset has been obtained for more testing.

A. PREDICTIONS WITH TEST SET 1

Using the *predict* function available in the Keras API, a confusion matrix has been produced along with test accuracy and loss scores for each model.



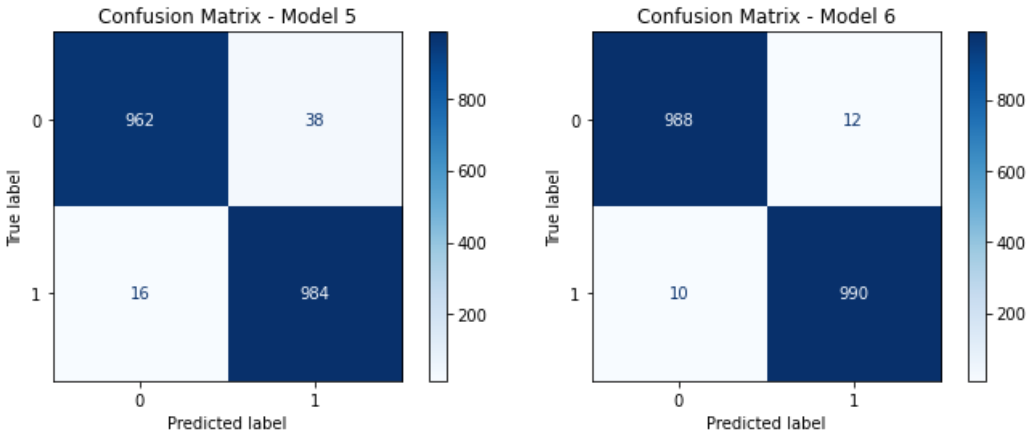


Figure 23: Confusion matrices for all six models. The antoniniani have been encoded as "1", whereas the denarii are represented by "0".

From the confusion matrices alone, it seems that the sixth model is the most efficient when it comes to predictions, although the fourth and second models have also produced significant results. The class “antoninianus” has been encoded as 1, and the class “denarius” as 0. The sixth model has made 10 false negative/type II errors (it has labelled 10 antoniniani as denarii) and 12 false positive/type I errors (it has labelled 12 denarii as antoniniani). The fourth model has only made 1 more type II error and 5 more type I errors compared to the sixth model; the second model has made 13 type II errors and 21 type I errors. The table below summaries all the metrics computed so far for every model:

Model	**Validation Loss**	**Validation Accuracy**	**Test Loss**	**Test Accuracy**
Baseline grayscale (1)	0.1015	0.9741	0.1454	0.9625
Updated grayscale (2)	0.0392	0.9887	0.0504	0.9830
Baseline RGB (3)	0.0573	0.9823	0.0720	0.9745
Updated RGB (4)	0.0379	0.9897	0.0473	0.9860
Baseline InceptionV3 (5)	0.0731	0.9766	0.0878	0.9730
Updated InceptionV3 (6)	**0.0322**	**0.9903**	**0.0376**	**0.9890**

Figure 24: Summary metrics for all six models. In bold are the “best” metrics (lowest loss and highest accuracy).

At this stage of the study, we can assume that the updated pre-trained InceptionV3 model is the most efficient one out of the six models built for this project, closely followed by the second and fourth models.

B. Grad-CAM VISUALISATIONS

To understand why the models have classified the two different kinds of denominations correctly- or incorrectly- the Grad-CAM algorithm [49] has been implemented to visualise the decision-making process of the models. The *tf-explain* package [40] has been used for this task, as it facilitates

the computation of the visualisations. The Grad-CAM algorithm has been used on mislabelled coins, as well as on the correctly identified ones.

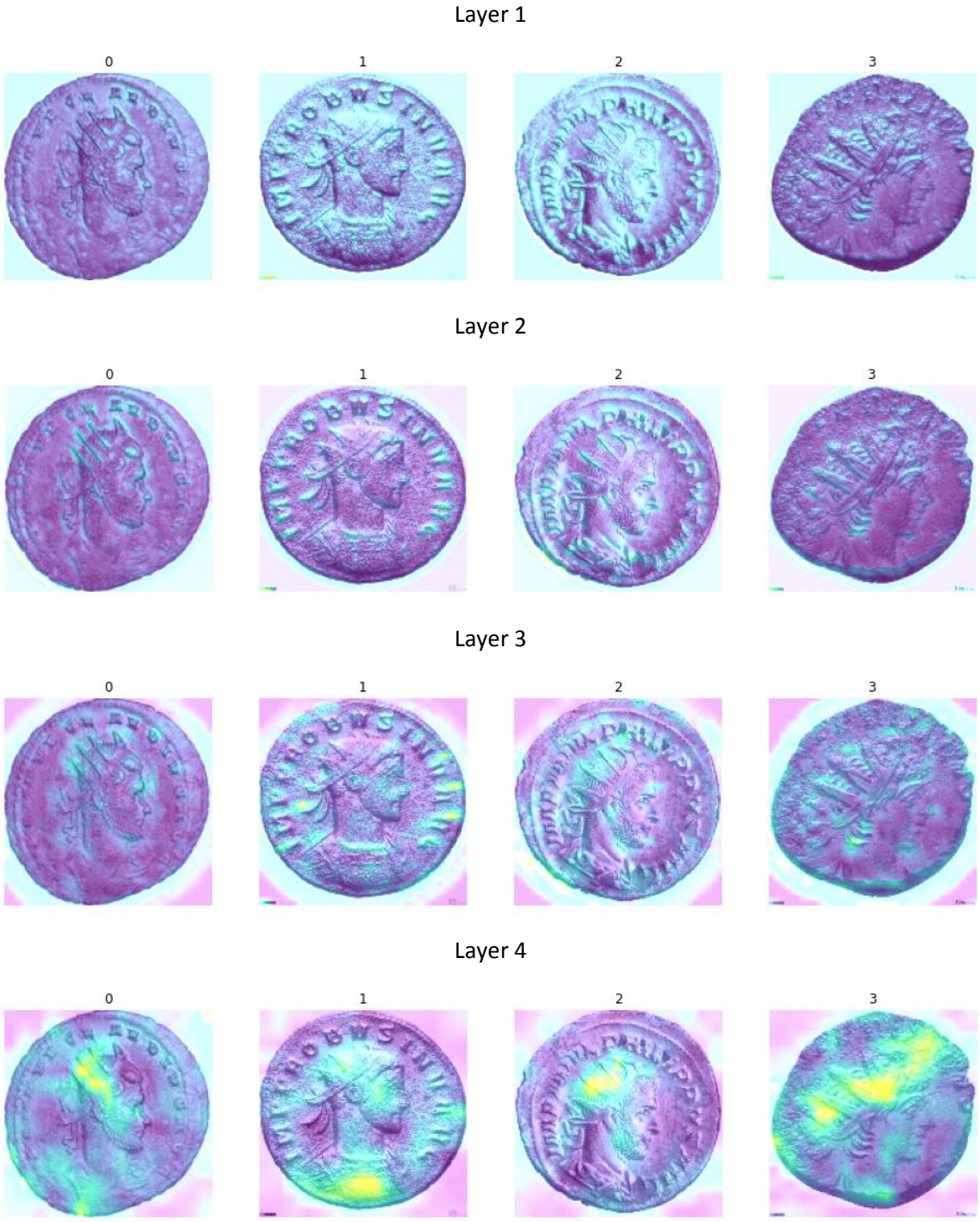


Figure 25: Gradual Grad-CAM visualisations per layer for correctly labelled antoniniani – Model 2

Gradient-weighted Class Activation Mapping (Grad-CAM) is an algorithm that produces heat maps, thus visually explaining how and which features the model learns to classify images. Using

gradients through each convolutional layer until the last one, it localises the features that determine the classification. The examples below have been produced by feeding the test set into the second model to highlight how it roughly highlights the edges during the first convolutions, then progressively gets more precise and focuses on more specific elements.

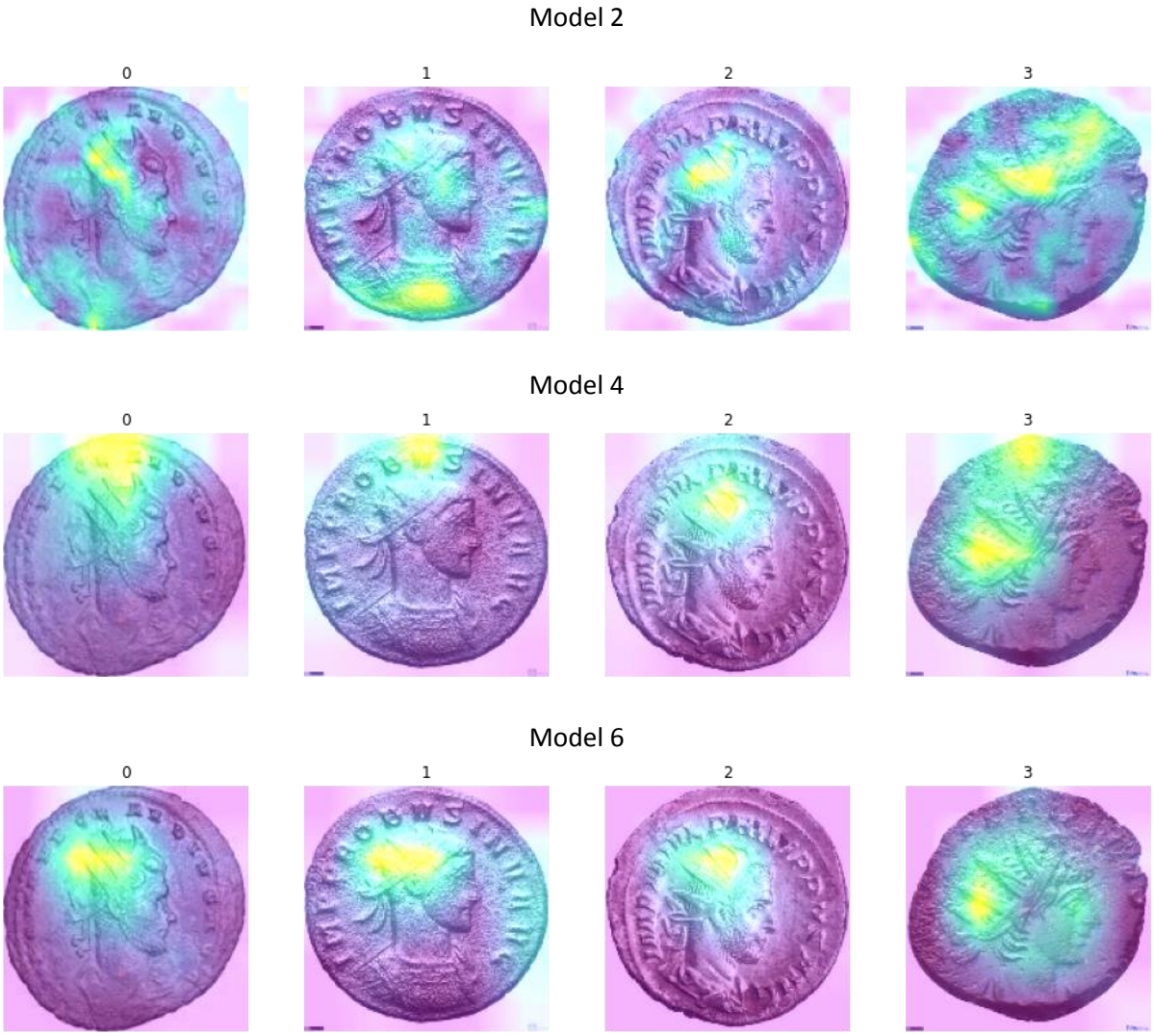


Figure 26: Grad-CAM of four examples correctly labelled by Model 2, 4 and 6.

Four examples of correctly labelled antoniniani by Model 2, 4 and 6 have been extracted from the test set. Using the Grad-CAM algorithm on the last convolutional layer of each model, it is interesting to note that the three models did not necessarily focus on the same features to identify the antoniniani. Apart from Model 6, which seemed to have focused solely on the radiate crown (at least for the four examples presented), the other two models have sometimes highlighted different parts of the coins to identify them. For instance, Model 2 has correctly identified example 1 using the bust of the emperor and not his crown. Model 4 seems to focus on a point at the top of the coins rather than the crown, visible on examples 0, 1, and 3, which might be an attempt at reading the lettering of the legend since it does evolve throughout the centuries.

Therefore, Grad-CAM is a valuable tool that brings necessary insights into the decisions made by the models: although this dissertation assumed that the models would discriminate the coins

according to the hairstyle of the portraits, which seemed to be the most obvious way to do so, the models seem to have found alternative ways to proceed, perhaps because the crowns' depictions are very varied and the coins are sometimes too damaged to be correctly read.

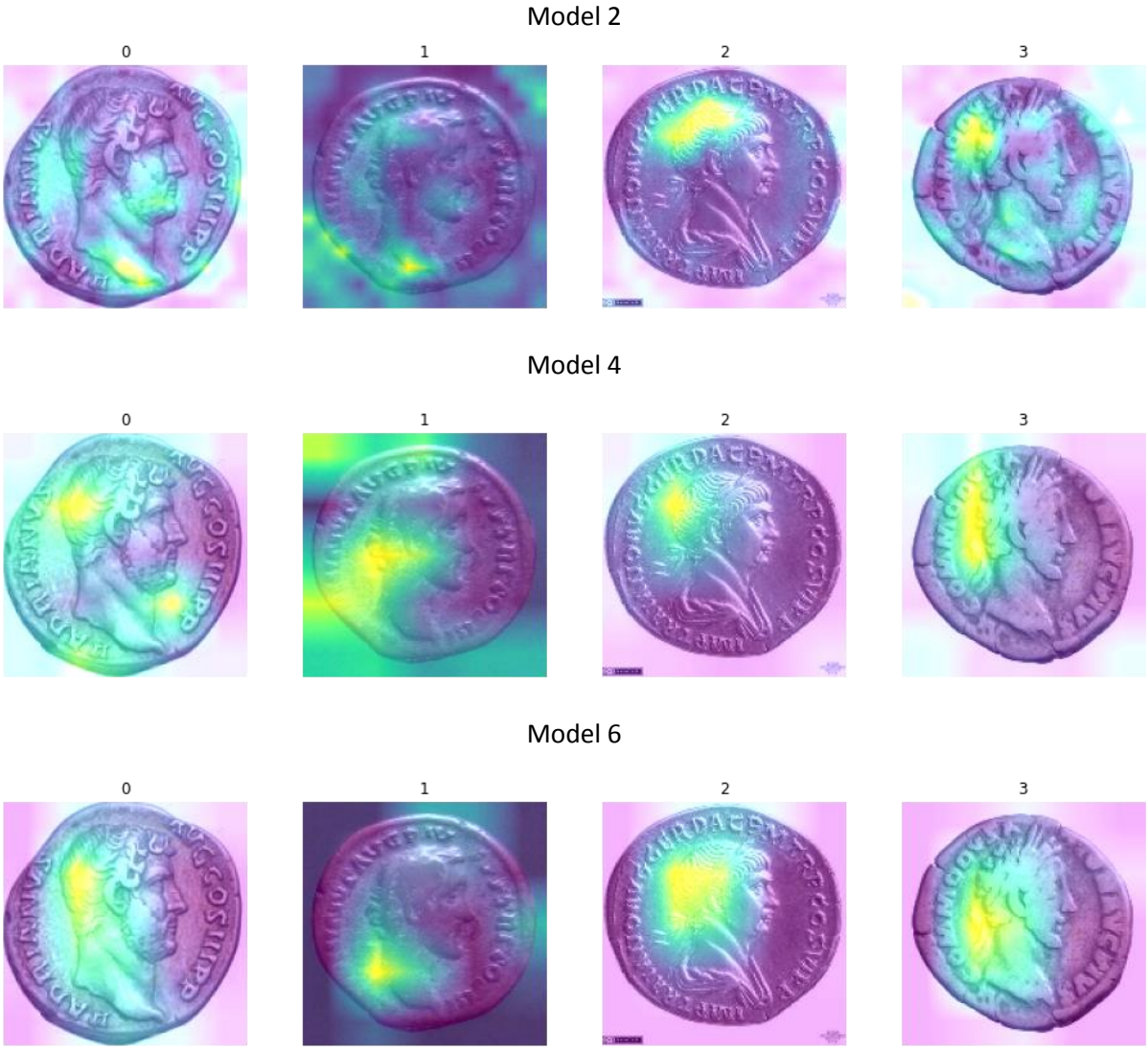


Figure 27: Grad-CAM examples of correctly identified denarii by Model 2, 4 and 6.

As for the denarii, the Grad-CAM has mainly highlighted the back of the head or neck of the emperors: instead of trying to recognise wreaths or bare heads, it seems the models have tried to identify the absence of a radiate crown, which is a sensible way to proceed. More examples of Grad-CAM visualisations of antoniniani and denarii are displayed in the appendices section.

The Grad-CAM algorithm is also helpful in spotting weaknesses and biases in the models and the dataset. Unfortunately, the models seem to consider the background and the tags on the images. This can be seen in the third example for Model 2 and 4. The green background of example 1 is highlighted for Model 4. This is a concerning issue since it means the models might not be able to be as efficient with another dataset since they are getting "hints" from tags and backgrounds. In other words, their ability to predict new data might be lower than anticipated.

As for the coins that the models have mislabelled, explanations are not always straightforward, even with the use of Grad-CAM. The Grad-CAM of every mislabelled coin is displayed in the appendices, but globally, it seems that the models have mislabelled coins because of several reasons:

- some coins are too worn out to be read;
- the tags and the green backgrounds seem to have confused the models;
- one bareheaded emperor was represented on an antoninianus and a radiate crown on a denarius (but not described as such in the tabular dataset);
- the radiate crowns are too thin and are hard to discern from the hair;
- In several instances, the models have decided to focus on the bare neck of the portrait instead of the head, ignoring the radiate crown and misclassifying it as denarius;
- It seems that the sixth model has misclassified two denarii because the emperors wore beards.

Finally, we can go back to the two denarii representing Augustus with a radiate crown, which were in the validation set during the training of the models and presented in the exploratory analysis. Below are displayed their Grad-CAM heatmaps after their images have been fed into Model 6. We can see that even though they both display noticeable crowns, the model has learnt to focus on other features to classify them correctly: the bare neck of Augustus – which might be why the models often inspect this feature as we have previously seen – a piece of hair from behind the head and a point on the top of the left coin, perhaps part of the legend.

Figure 28: Grad-CAM of two denarii representing Augustus with a radiate crown.

Following up on the insights provided by the Grad-CAM visualisations and concerns regarding the biases introduced by tags and backgrounds, the models have been tested again on a different dataset.

C. PREDICTIONS WITH TEST SET 2

A new dataset of 8,000 coins (4,000 antoniniani and 4,000 denarii) has been kindly provided by Ognjen Arandjelovic [10]. The images come from various auction houses; they have been carefully sorted so they would not contain any tag and display the same white background.

Figure 29: Examples extracted from the second test set.

To verify the uniformity (same background, no tags) of the dataset, the pixels of the images have been averaged again for each denomination. The background is the same for both denominations; no tags are visible.

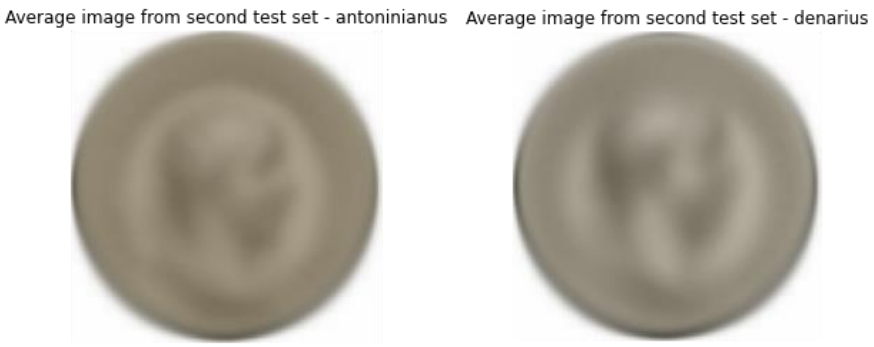


Figure 30: Averaged images of the second test set of each denomination.

Here are the confusion matrices produced after predictions have been computed for every model:

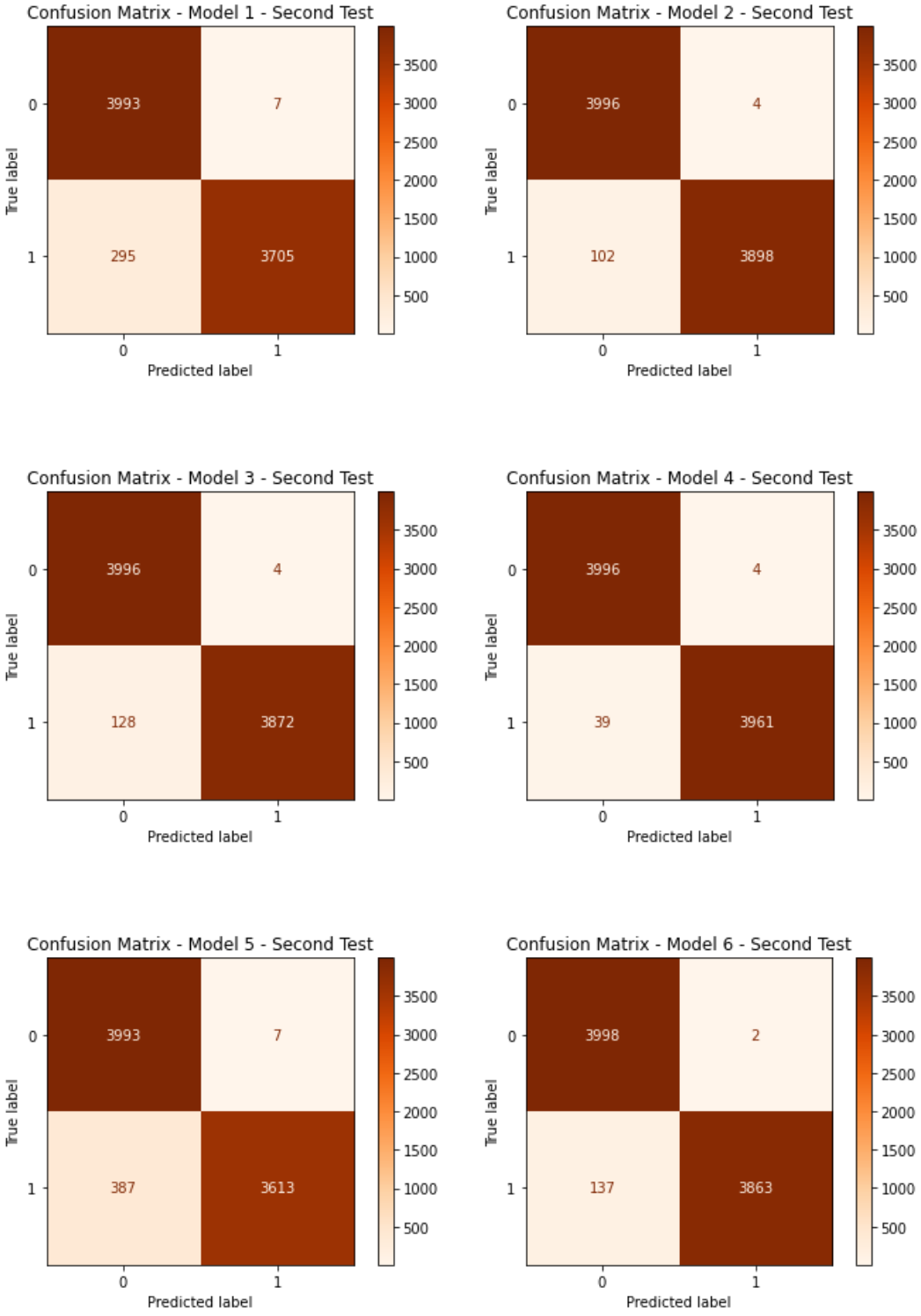


Figure 31: Confusion matrices for every model (second test). "1" stands for antoninianus, "0" for denarius.

This time, it is the fourth model which scored the best accuracy (0.9946), followed by the second model (0.9868), then the baseline RGB CNN (0.9835). The sixth model (the fine-tuned InceptionV3 model) is only in the fourth position regarding accuracy for the second test, whereas its score was 0.9890 during the first test. These numbers are very close to each other, and variations can be due to the stochastic nature of these processes. However, they might also indicate that the initial

dataset was slightly biased due to tagging and differences in backgrounds, or perhaps noise detected by the sixth model that is not present in the new dataset.

It is interesting to note there are way more type II errors for this dataset than type I errors for all models: all networks, regardless of their predictive abilities, seem to have more trouble labelling antoniniani as such and tend to mislabel them as denarii.

Model	**Val. Loss**	**Val. Acc.**	**Test Loss**	**Test Acc.**	**Test 2 Loss**	**Test 2 Acc.**
Baseline grayscale (1)	0.1015	0.9741	0.1454	0.9625	0.1421	0.9622
Updated grayscale (2)	0.0392	0.9887	0.0504	0.9830	0.0391	0.9868
Baseline RGB (3)	0.0573	0.9823	0.0720	0.9745	0.0705	0.9835
Updated RGB (4)	0.0379	0.9897	0.0473	0.9860	**0.0205**	**0.9946**
Baseline InceptionV3 (5)	0.0731	0.9766	0.0879	0.9730	0.1779	0.9507
Updated InceptionV3 (6)	**0.0322**	**0.9903**	**0.0376**	**0.9890**	0.0611	0.9826

Figure 32: Summary of all metrics computed for the six models. In bold are the "best" metrics (lowest loss and highest accuracy).

The Grad-CAM of the correctly labelled coins, presented in this appendices section of this study, shows similar patterns during the testing of the original dataset: the recognition of the radiate crowns on the antoniniani, with sometimes parts of the legends, facial hair, or the bust highlighted on the heatmaps.

As for the denarii, the models tend to focus on the back of the neck and the head, most probably to recognise the absence of a radiate crown, and sometimes parts of the legend and facial hair.

Interestingly, the Grad-CAM of the denarii correctly predicted by Model 6 often highlights the entire neck of the portrait, which was not the case with the previous dataset.

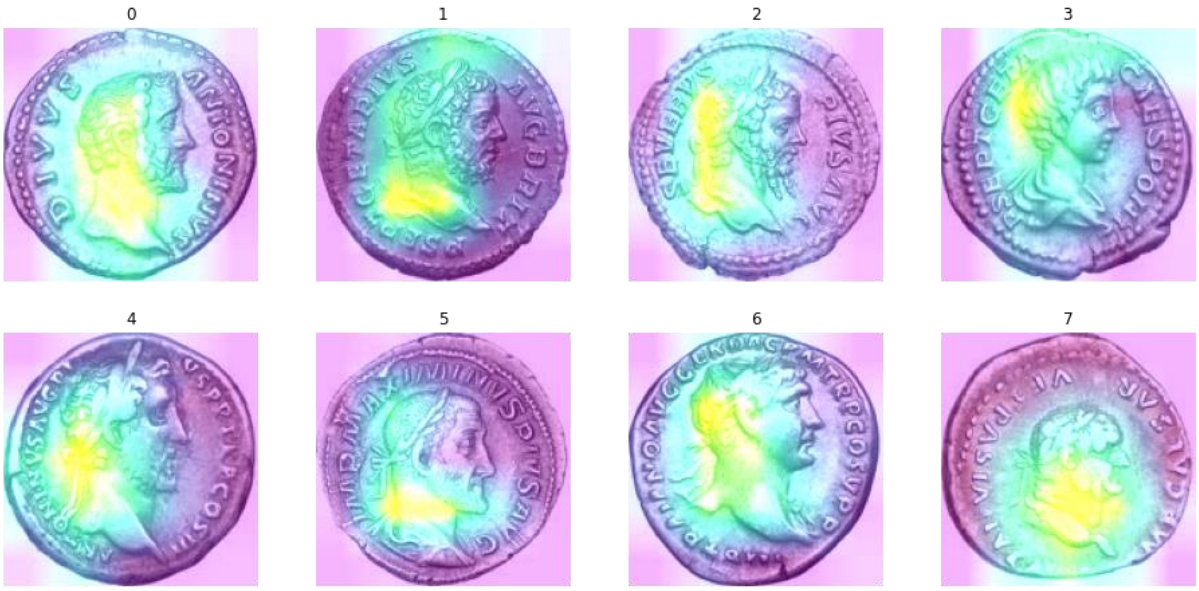


Figure 33: Grad-CAM of denarii correctly labelled by Model 6 - Second Test.

This might explain why the number of type II errors is higher for Model 6. Indeed, several examples of the mislabelled antoniniani show that the model ignores the crown to focus instead on the emperor's neck. A similar phenomenon can be noticed in the mislabelled antoniniani of Model 4.

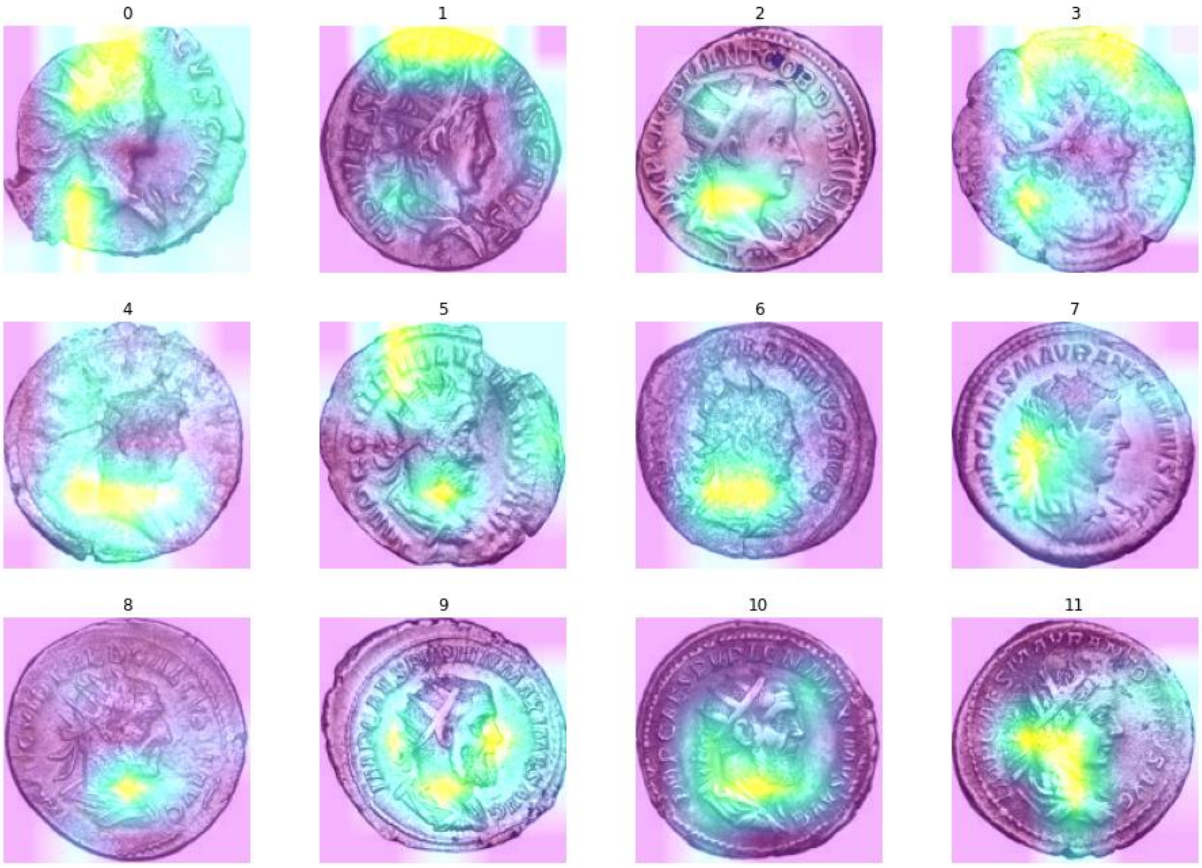


Figure 34: Grad-CAM of mislabelled antoniniani by Model 6 - Second Test.

6. DISCUSSION

The purpose of this study was to allow numismatic professionals to classify their coin collections more efficiently. Indeed, following the acceleration of the digitization of objects intended for exhibition or sale, new tools are necessary, and the automation of these processes seems more and more necessary.

More particularly, the classification of ancient coins is a real challenge: their handmade minting and state of conservation often hinder their identification. In this context, computer vision can provide valuable assistance, particularly the promising performances shown by neural networks.

This dissertation focused on the classification of Roman imperial coins: more particularly, its purpose was to use deep learning as a way to distinguish the denarius from the antoninianus, the latter (almost) systematically representing the reigning emperor wearing a radiate crown on the obverse, while the representations on the obverse of the denarius are more varied (wreaths, helmets, bare heads).

For this task, three convolutional neural networks have been built, from three existing architectures used as baseline methods, with the assumption of training them to recognize the presence or absence of radiate crown on a coin, from a dataset of 23,000 examples: a convolutional neural network with four layers trained on grayscale images; a convolutional neural network of 5 layers trained on colour images; and finally, a pre-trained InceptionV3 model.

The three models all produced significant results, with 0.9830, 0.9860, and 0.9890 of test accuracy for Model 2, 4 and 6, respectively. The InceptionV3 model was the most efficient during the first test, but the difference between the three accuracy scores is relatively small.

The use of the Grad-CAM algorithm, applied to the (correct and incorrect) predictions of the three models, revealed a potential bias: the use of logos by the institutions and the differences in background colours behind the pictured coins.

To overcome this problem and verify the actual performance of the models, they were tested again on a new dataset; it has been cleaned of all tags and backgrounds different from white.

This second test revealed that model 4 was the most efficient network. Furthermore, the models made many type II errors: a tendency to mislabel antoniniani as denarii has been detected. Looking at the Grad-CAM visualisations in more detail, Model 6 often focused on the bare neck of the portrait while ignoring the hairstyle entirely.

These errors might be due to the decision to keep (and unexpectedly find) "divergent" coins, such as antoniniani without any radiate crown or denarii depicting it in the dataset. Along with museum logos and varied backgrounds, this might have confused the models and forced them to search for alternative discriminative features.

However, the accuracy of the three models for this second test remains significant: 0.9868 for model 2, 0.9946 for model 4, and 0.9826 for model 6.

Ultimately, this study showed that the models classified the coins mainly according to hairstyles, confirming this project's assumption. However, this conclusion needs to be tempered by elements that were not initially foreseen. Beyond the potential biases introduced by museum logos and differences in background colours, it seems the models have sometimes learnt unexpected features leading to a correct classification (for example, parts of the legends, details on the bust, or the face of the emperors). In other words, they seem to have exceeded the (modest) domain knowledge on which they were based.

REFERENCES

1. https://www.washingtonpost.com/business/capitalbusiness/the-smithsonian-turned-to-conveyors-belts-cameras-to-digitize-its-many-artifacts/2015/01/22/6d39e9b2-9db1-11e4-a7ee-526210d665b4_story.html
2. https://nomisma.org
3. Bruun P. (1999) "Imperial Government" in *Roman coins and public life under the empire: E. Togo Salmon Papers II*, *2*, p. 19.
4. Wilson A. I. (2007) "The metal supply of the Roman Empire" in *Supplying Roman and the Empire*, edited by Papi E., Journal of Roman Archaeology Supplementary Series Number 69, p.109.
5. Pense A. W. (1992) "The decline and fall of the roman denarius" in *Materials characterization*, *29* (2), p. 213-222.
6. Mattingly, H. (1932) "Hoards of Roman coins found in Britain and a coin survey of the Roman province" in *The Journal of Roman Studies*, *22*(1), p. 88-95.
7. Howgego, C. (2009) "Some Numismatic Approaches to Quantifying the Roman Economy" in *Quantifying the Roman Economy* edited by Bowman A. and Wilson A., Oxford University Press, p. 287.
8. Oman C. (1916) "The decline and fall of the denarius in the third century A.D." in *The Numismatic Chronicle and Journal of the Royal Numismatic Society, Fourth Series, Vol. 16*, Royal Numismatic Society, pp. 37-60.
9. Levic, B. (1982) "Propaganda and the imperial coinage" in *Antichthon*, *16*, p.104-116.
10. Cooper J., Arandjelovic (2020) "Understanding Ancient Coin Images", Springer Nature Switzerland AG 2020 edited by Oneto L. et al., p.330.
11. Huber-Mörk R. et al. (2011) "Identification of ancient coins based on fusion of shape and local features", in *March. Vis. Appl. 22*, p. 983.
12. Kim J., Pavlovic V. (2016) "Discovering characteristic landmarks on ancient coins using convolutional networks" in *J. Electron. Imaging* 26(1).
13. Sutherland C. H. V. et al. (1984) *The Roman Imperial Coinage*, Vol. 1, Spink.
14. Krizhevsky A., Sutskever I., Hinton G.E. (2012) "ImageNet classification with deep convolutional neural networks" in *Advances in Neural Information Processing Systems*.
15. Kim J., Pavlovic V. (2014) "Improving ancient roman coin recognition with alignment and spatial encoding", in Proc. Workshop on European Computer Vision Conf. (ECCV).
16. Schlag I., Arandjelovic O. (2017) "Ancient roman coin recognition in the wild using deep learning base recognition of artistically depicted face profiles", in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, p. 2898.
17. Simonyan S., Zisserman A. (2014) "Very deep convolutional networks for large-scale image recognition", ArXiv, p. 1409.1556.
18. Pan X., Tougne L. (2018), "Image analysis and deep learning for aiding professional coin grading" in *Proc. SPIE 10836 2018 International Conference on Image and Video Processing and Artificial Intelligence 1083605.*
19. https://numismatics.org
20. http://nomisma.org/datasets
21. https://creativecommons.org/licenses/by/3.0/
22. https://www.smb.museum/museen-einrichtungen/muenzkabinett/home/
23. https://www.ashmolean.org/heberden-coin-room
24. https://medaillesetantiques.bnf.fr/ws/catalogue/app/report/index.html
25. https://www.britishmuseum.org/collection/galleries/money
26. http://nomisma.org/sparql/
27. Lusnia S. S. (1995) "Julia Domna's coinage and Severan dynastic propaganda" in *Latomus*, *54* (Fasc. 1), p.119.

28. De Callataÿ F. (1995) “Calculating ancient coin production: seeking a balance” in *The Numismatic Chronicle (1966-)*, p. 289.
29. Manders E. (2012) *Coining images of power: patterns in the representation of Roman emperors on imperial coinage, AD 193-284* (Vol. 15). Brill.
30. Steyn D. (2014) “Chasing the Sun: Coinage and Solar Worship in the Roman Empire of the Third and Early Fourth Centuries CE” in *Records of the Canterbury Museum Volume 28 2014*, *28*, p.31.
31. Denarius. (2022, September 16). In Wikipedia. https://en.wikipedia.org/wiki/Denarius.
32. http://gams.uni-graz.at/context:numis
33. Chollet F. (2021) *Deep Learning with Python: Second Edition*, Manning, p.7, 204, 69, 202, 48, 216.
34. https://www.tensorflow.org/
35. https://keras.io/
36. https://colab.research.google.com/
37. Morishita M et al. (2017) “An empirical study of mini-batch creation strategies for neural machine translation”, arXiv:1706.05765.
38. Moroney L. (2020) *AI and Machine Learning for Coders: A Programmer’s Guide to Artificial Intelligence*, O’Reilly, p.27, 34, 35, 26, 61, 64, 5.
39. Aggarwal C. C. (2018) *Neural Networks and Deep Learning: A Textbook,* Springer, p. 12-15, 138.
40. Meudec, R. (2021) “tf-explain: Version 0.3.1”, https://doi.org/10.5281/zenodo.5711704
41. https://github.com/sicara/tf-explain/issues/73
42. Zou F. et al. (2019) “A sufficient condition for convergences of adam and rmsprop” In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, p. 11127-11135.
43. Mukhoti J. et al. (2020) “Calibrating Deep Neural Networks using Focal Loss” in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*
44. Kingma D. P., Ba J. (2014) “Adam: A Method for Stochastic Optimization”, arXiv:1412.6980.
45. https://www.kaggle.com/c/dogs-vs-cats
46. Szegedy C. et al. (2014) “Going deeper with convolutions”, arXiv:1409.4842v1.
47. Szegedy C. et al. (2015) “Rethinking the Inception Architecture for Computer Vision”, arXiv:1512.00567v3
48. https://keras.io/api/applications/
49. Selvaraju R. R. (2019) “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, arXiv:1610.02391v4.

APPENDICES

APPENDIX I: MODEL 2 Grad-CAM VISUALISATIONS - TEST 1

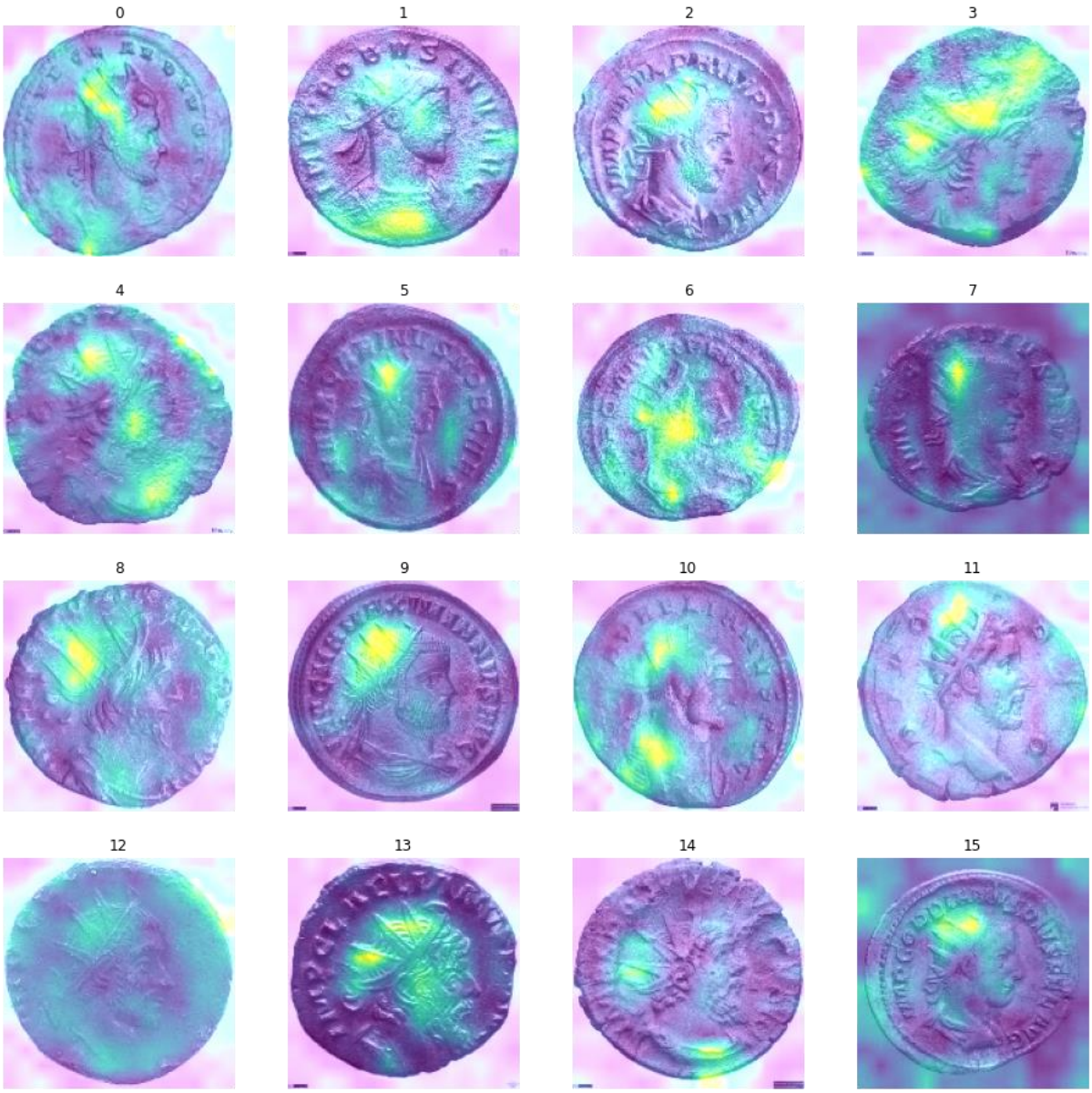


Figure 35: Grad-CAM of correctly labelled antoniniani - Model 2 – test 1

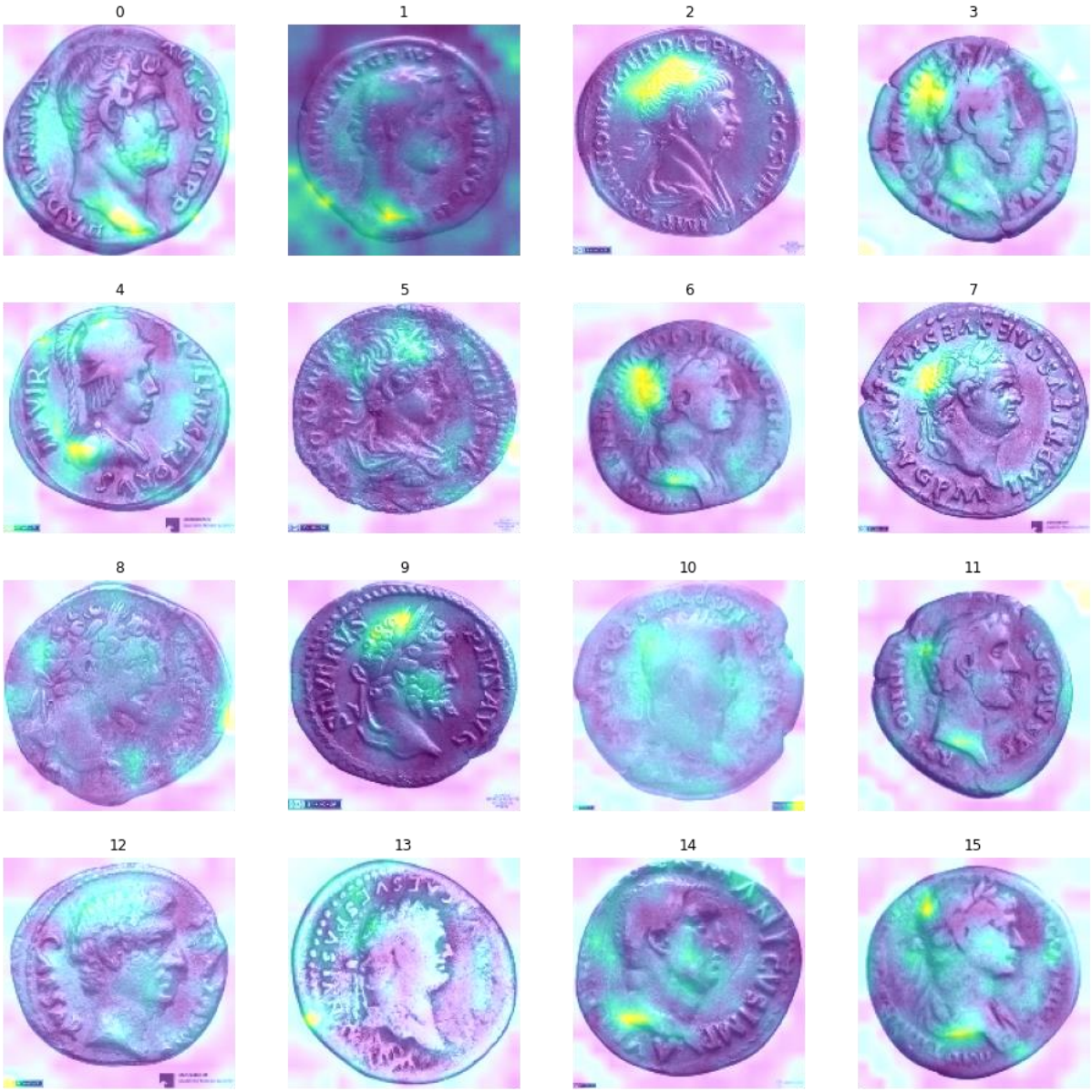


Figure 36: Grad-CAM of correctly labelled denarii - Model 2 - Test 1.

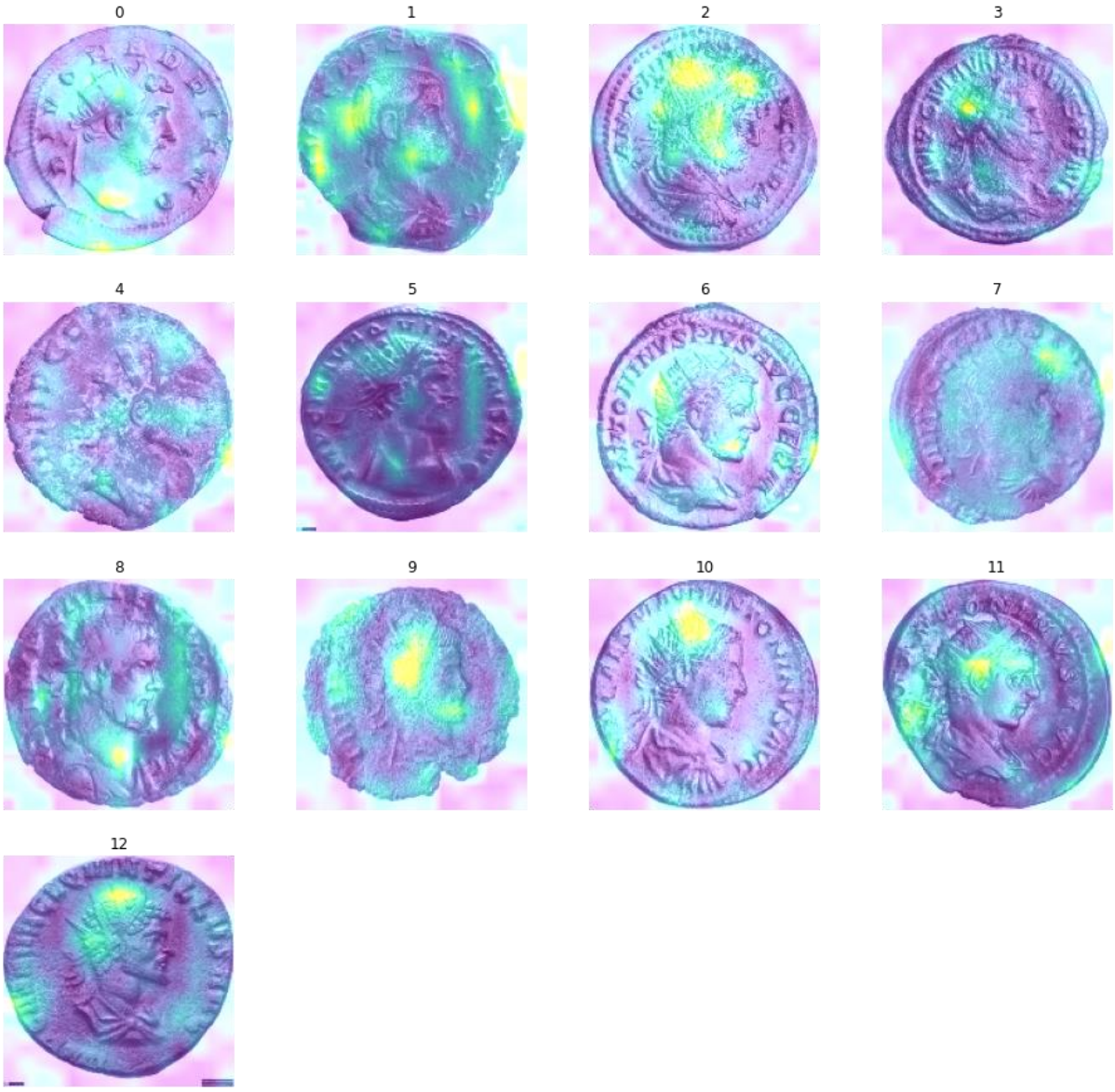


Figure 37: Grad-CAM of mislabelled antoniniani - Model 2 - Test 1

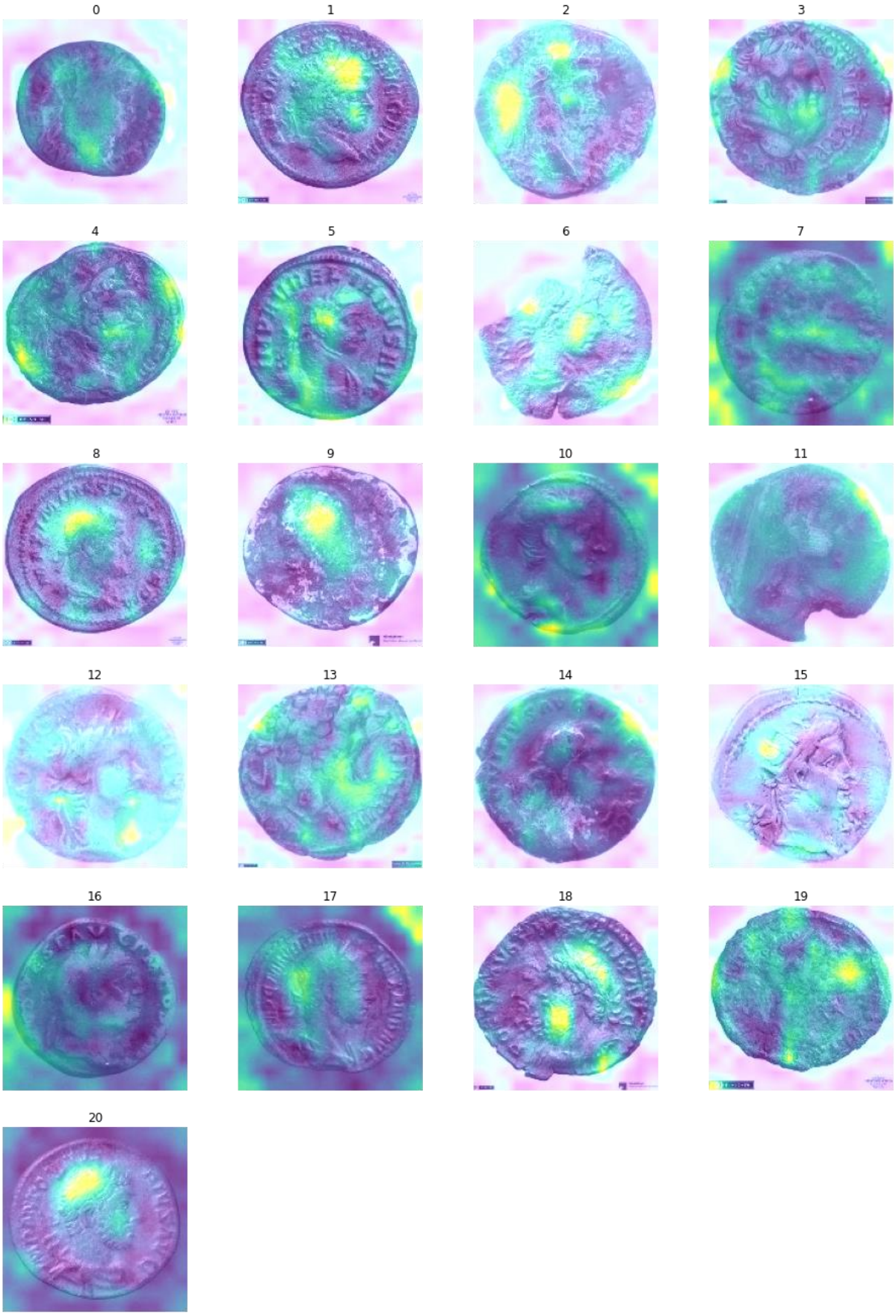


Figure 38: Grad-CAM of mislabelled denarii - Model 2 - Test 1.

APPENDIX II: MODEL 2 Grad-CAM VISUALISATIONS - TEST 2

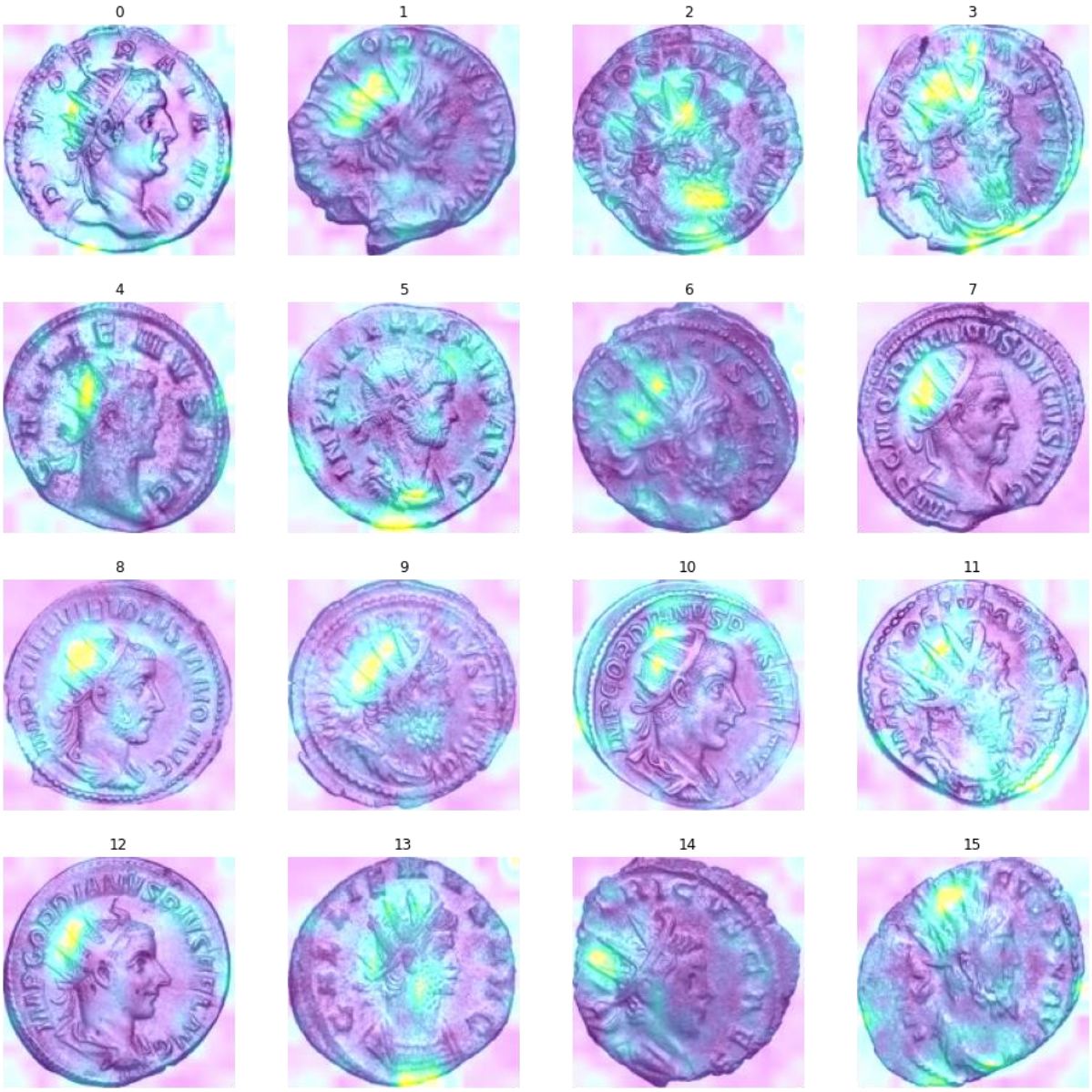


Figure 39: Grad-CAM of correctly labelled antoniniani - Model 2 - Test 2.

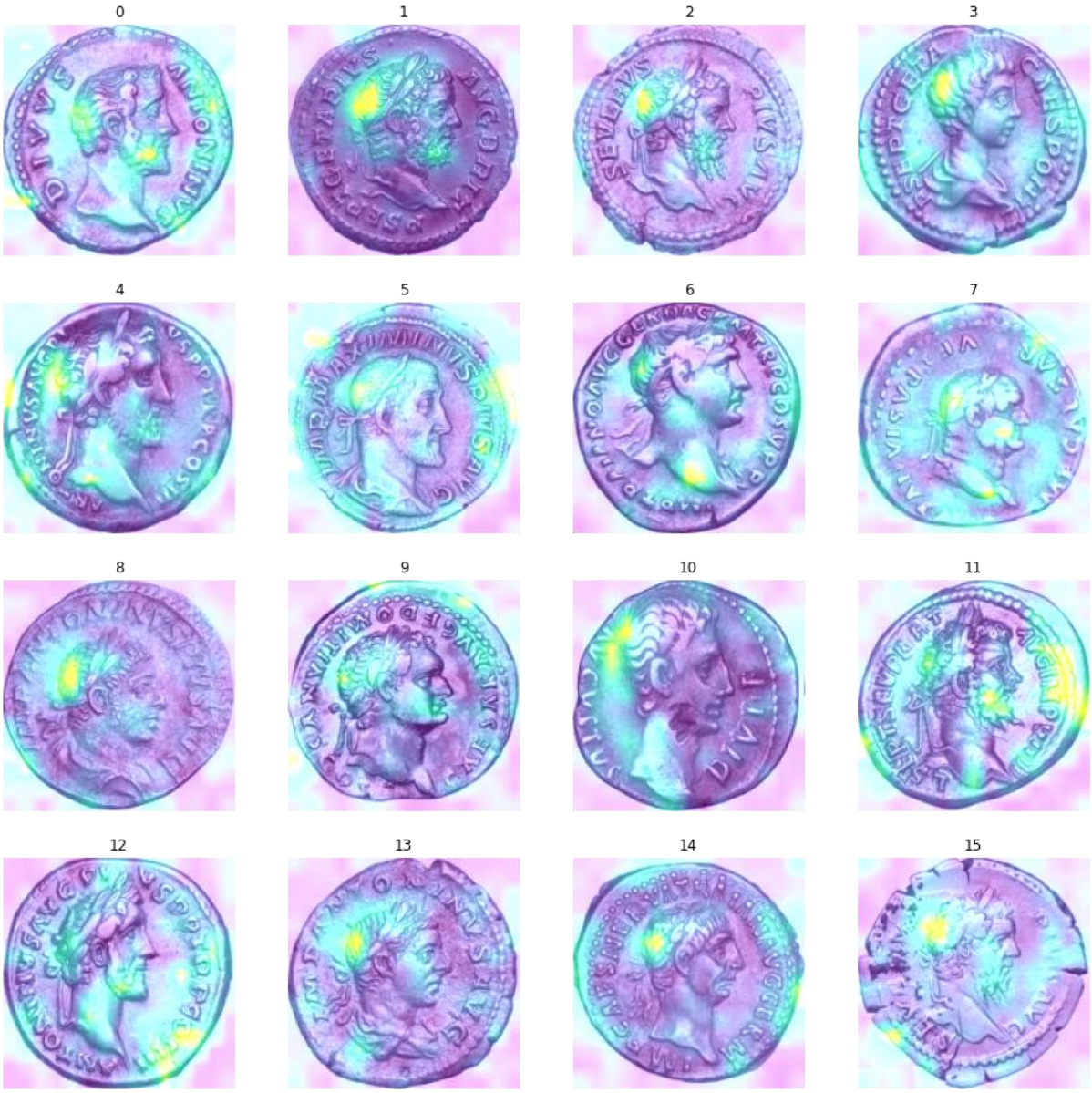


Figure 40: Grad-CAM of correctly labelled denarii - Model 2 - Test 2.

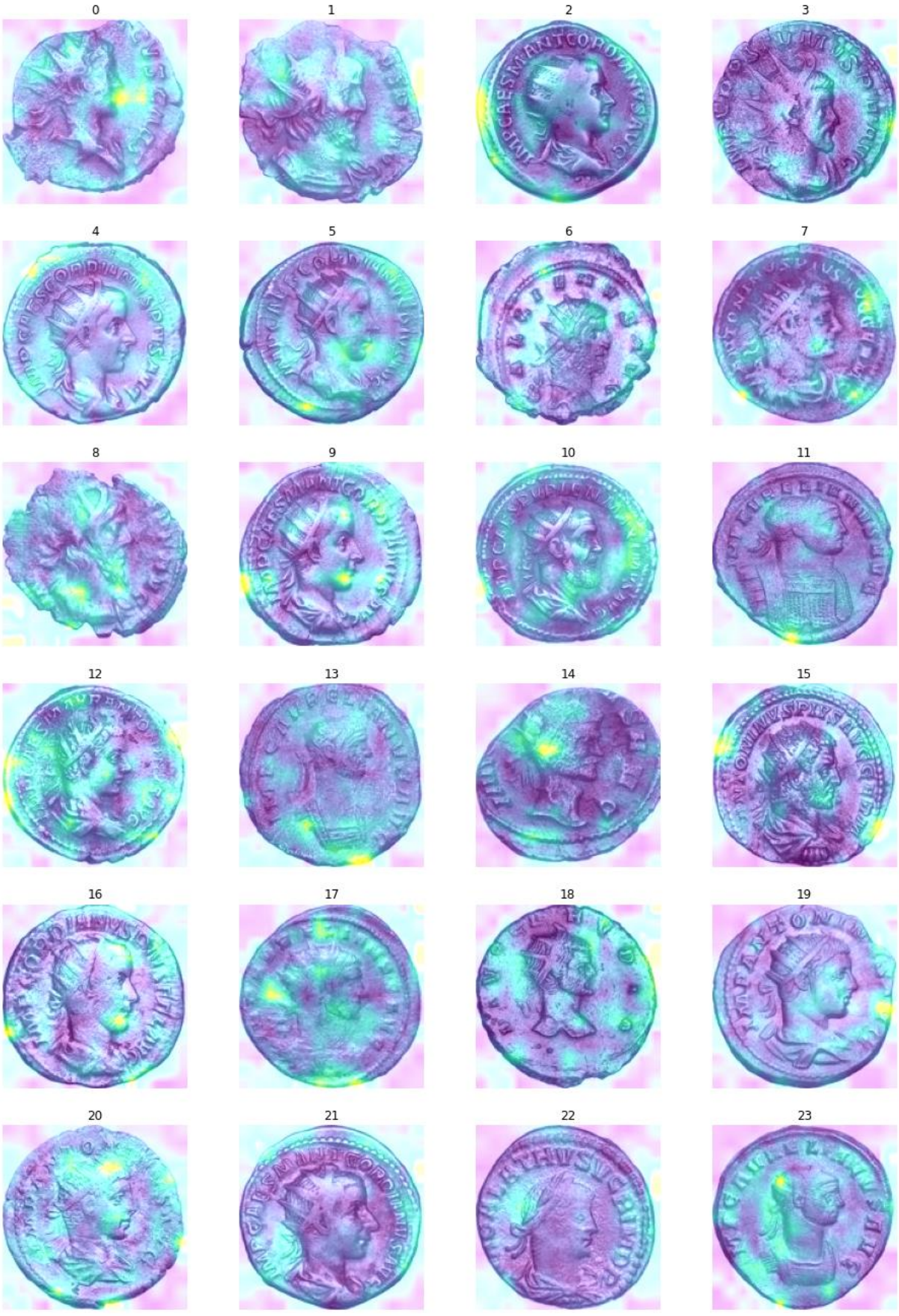


Figure 41:Grad-CAM of mislabelled antoniniani - Model 2 - Test 2 (excerpt).

Figure 42: Grad-CAM of mislabelled denarii - Model 2 - Test 2.

APPENDIX III: MODEL 4 Grad-CAM VISUALISATIONS - TEST 1

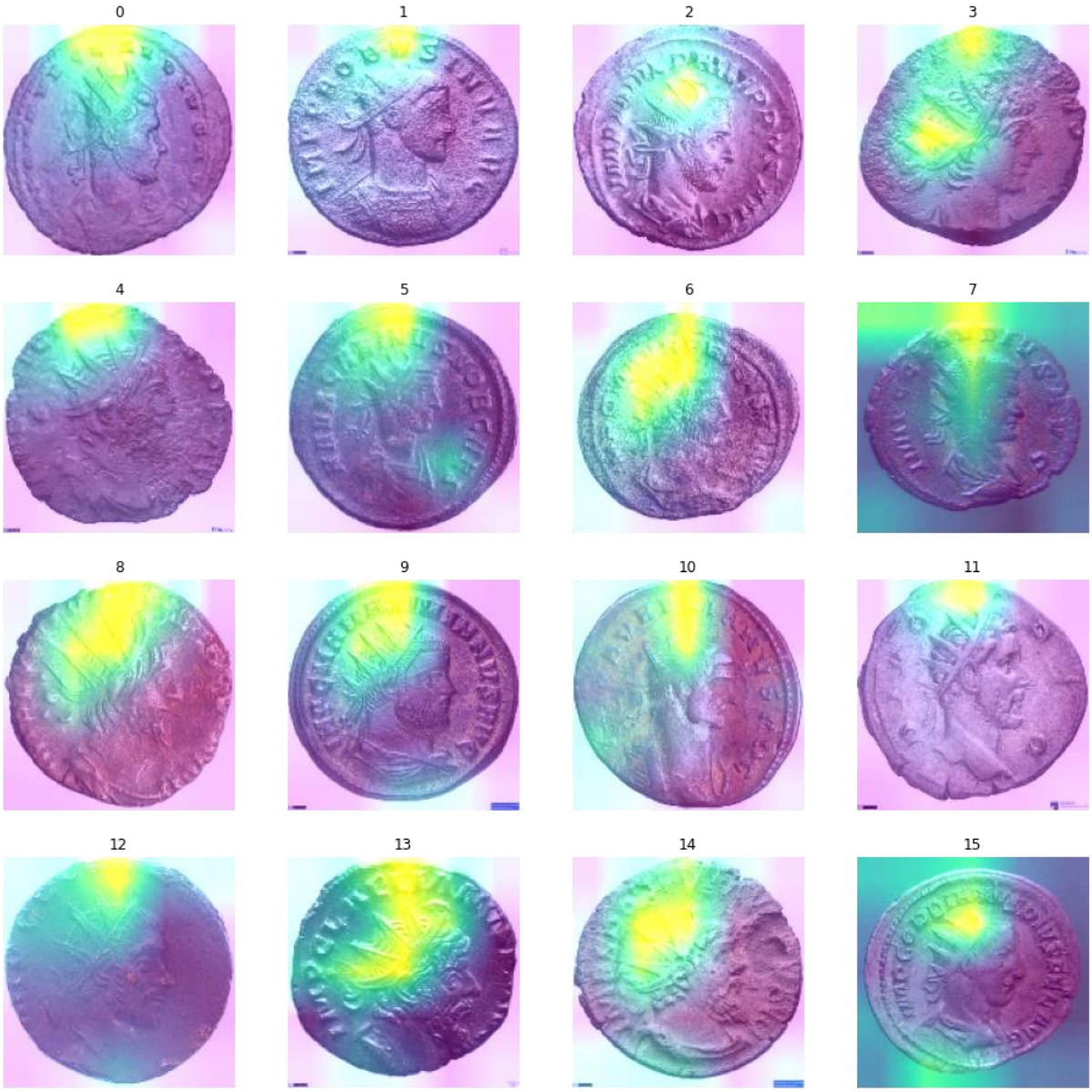


Figure 43: Grad-CAM of correctly labelled antoniniani - Model 4 - Test 1.

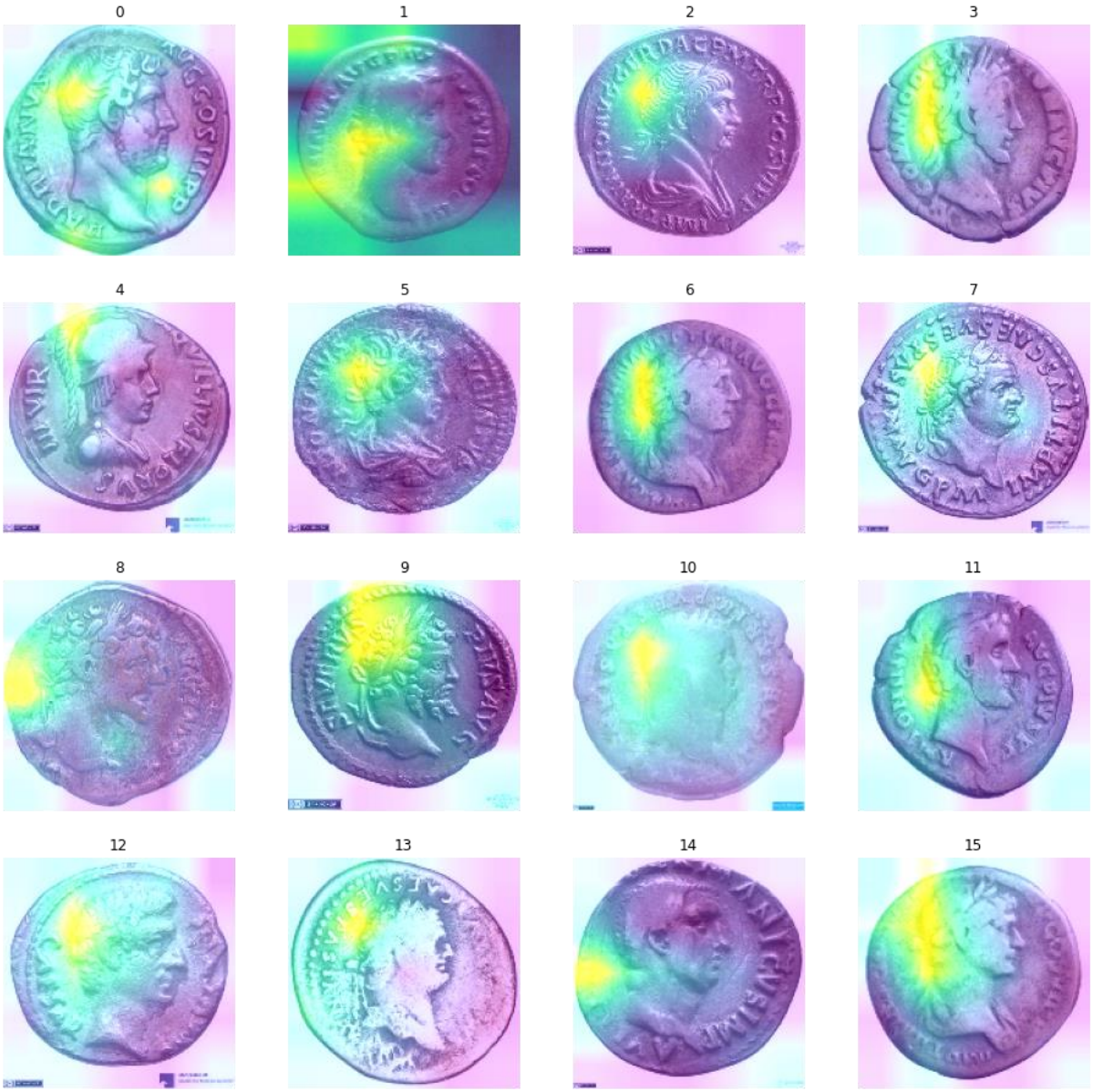


Figure 44: Grad-CAM of correctly labelled denarii - Model 4 - Test 1.

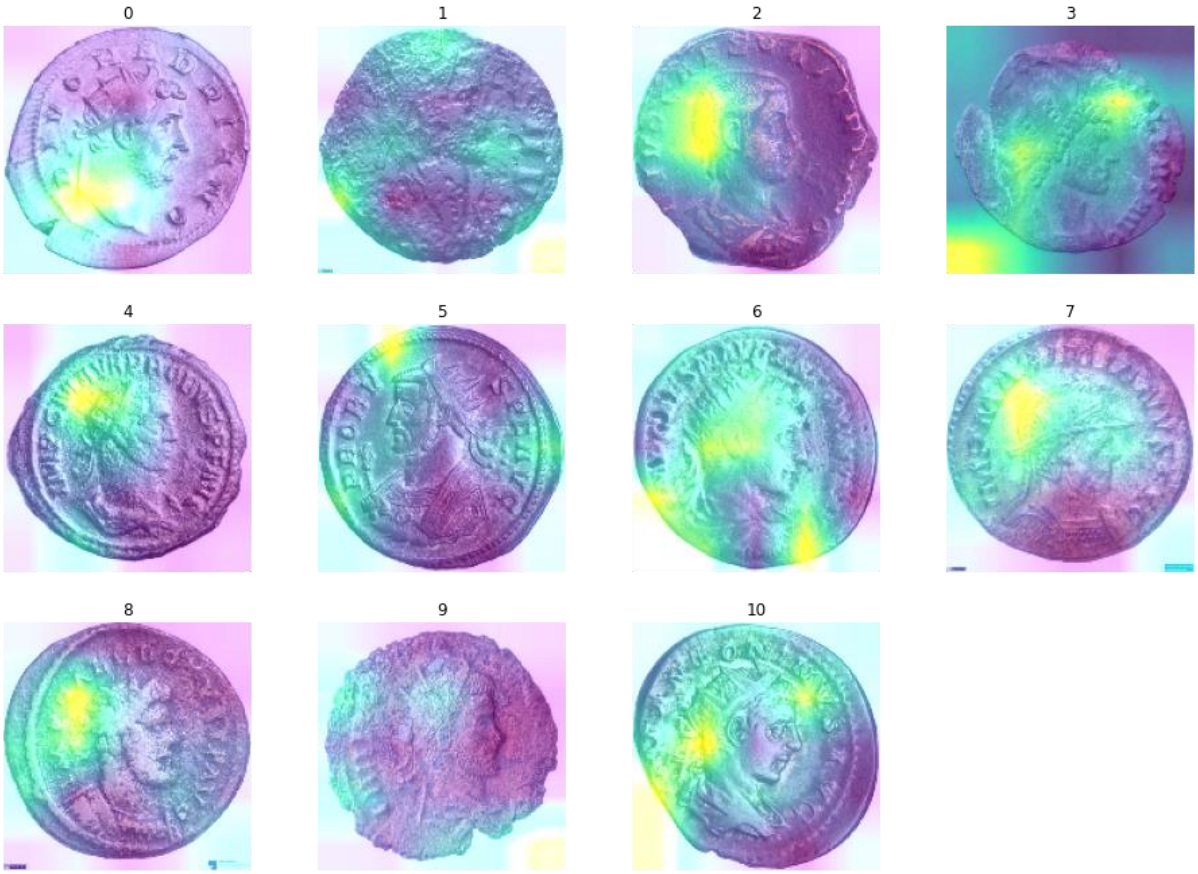


Figure 45: Grad-CAM of mislabelled antoniniani - Model 4 - Test 1.

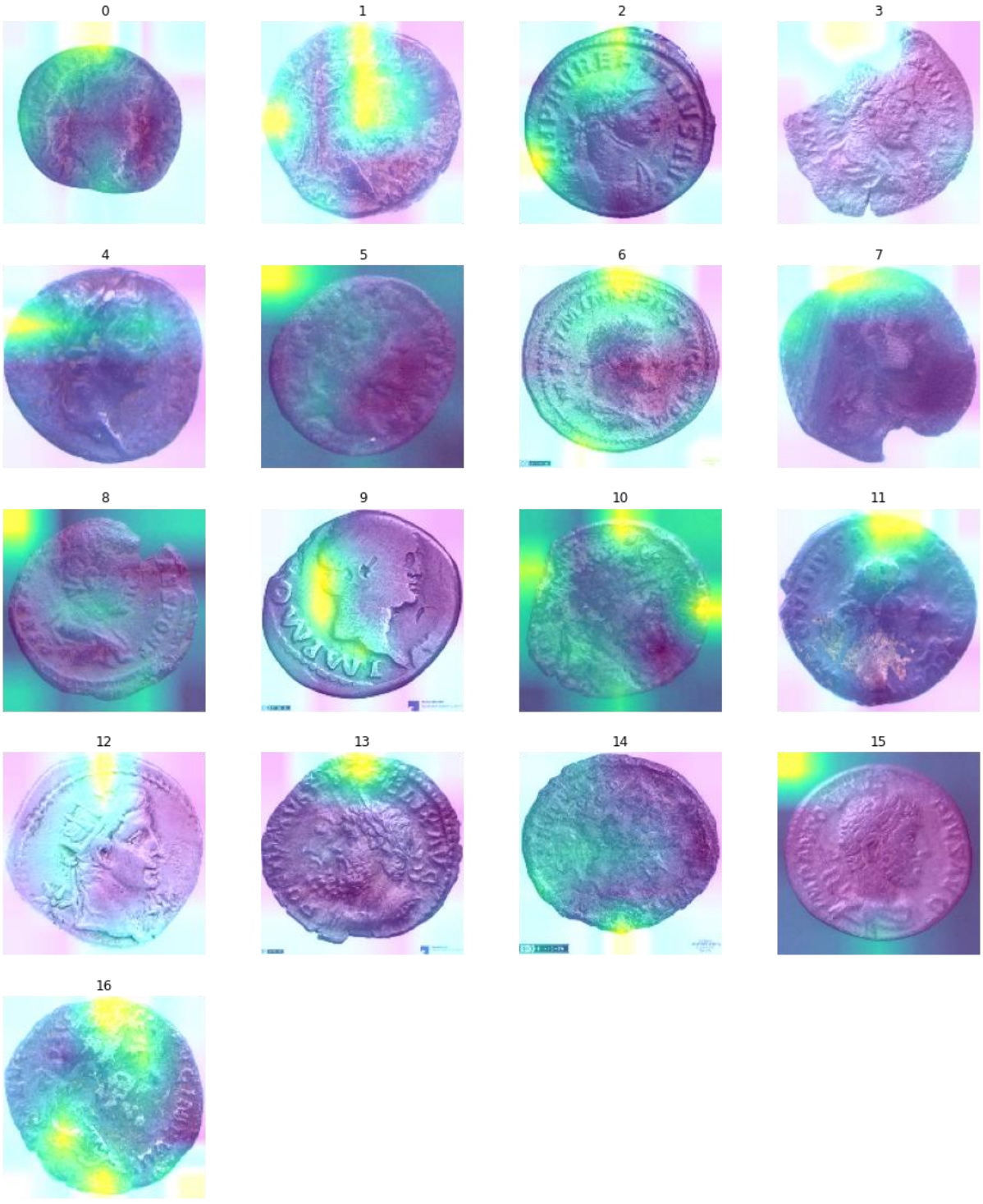


Figure 46: Grad-CAM of mislabelled denarii - Model 4 - Test 1.

APPENDIX IV: MODEL 4 Grad-CAM VISUALISATIONS - TEST 2

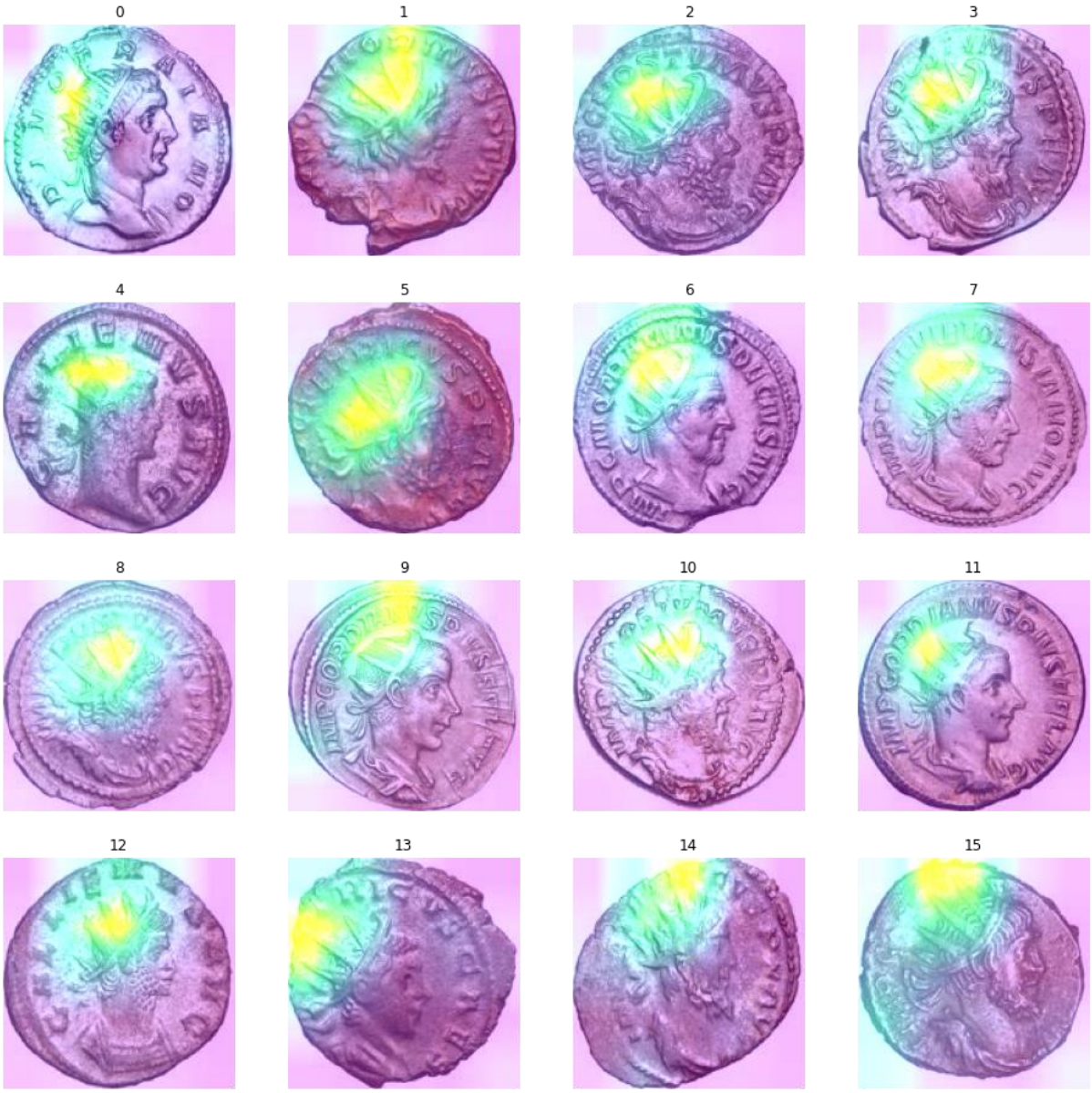


Figure 47: Grad-CAM of correctly labelled antoniniani - Model 4 - Test 2.

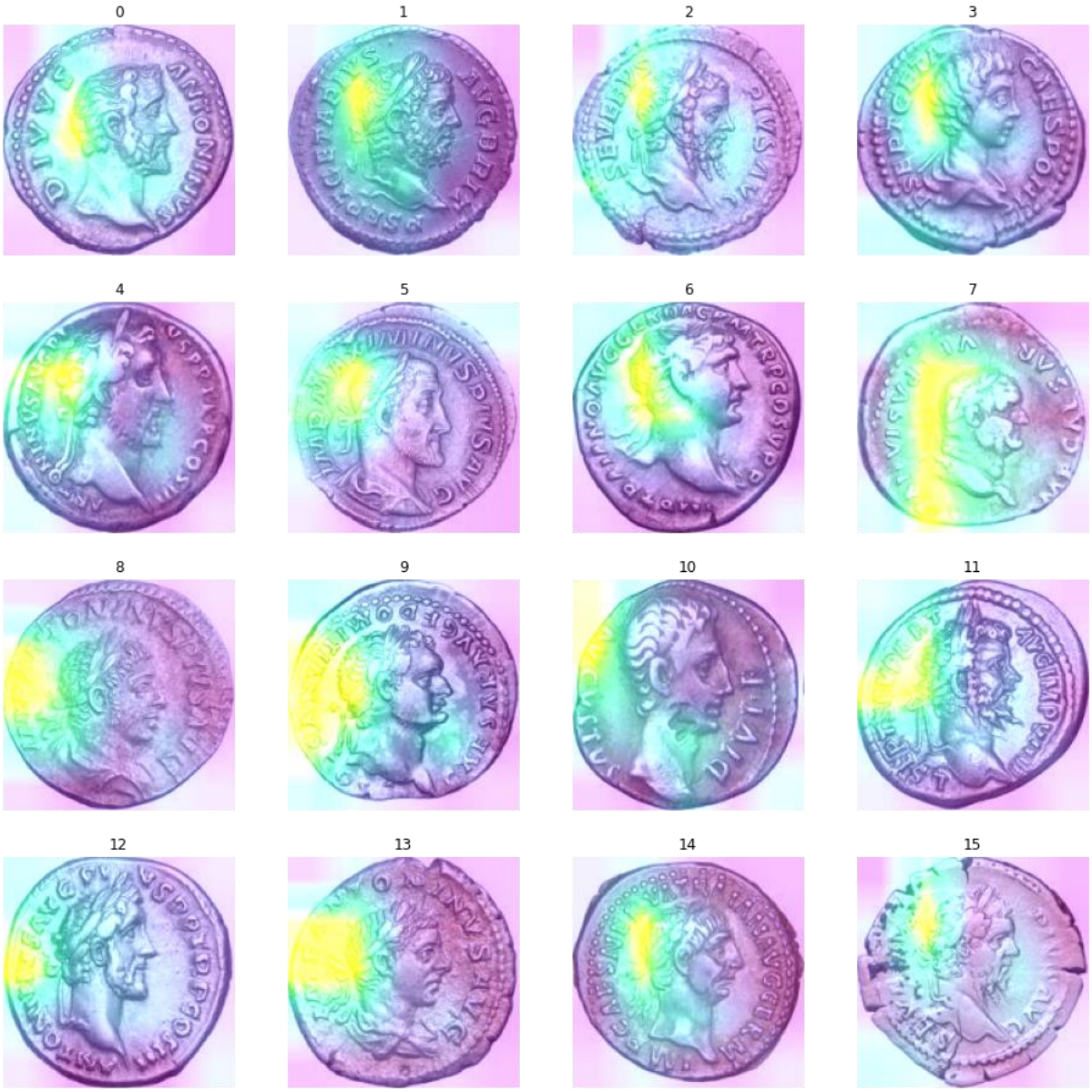


Figure 48: Grad-CAM of correctly labelled denarii - Model 4 - Test 2.

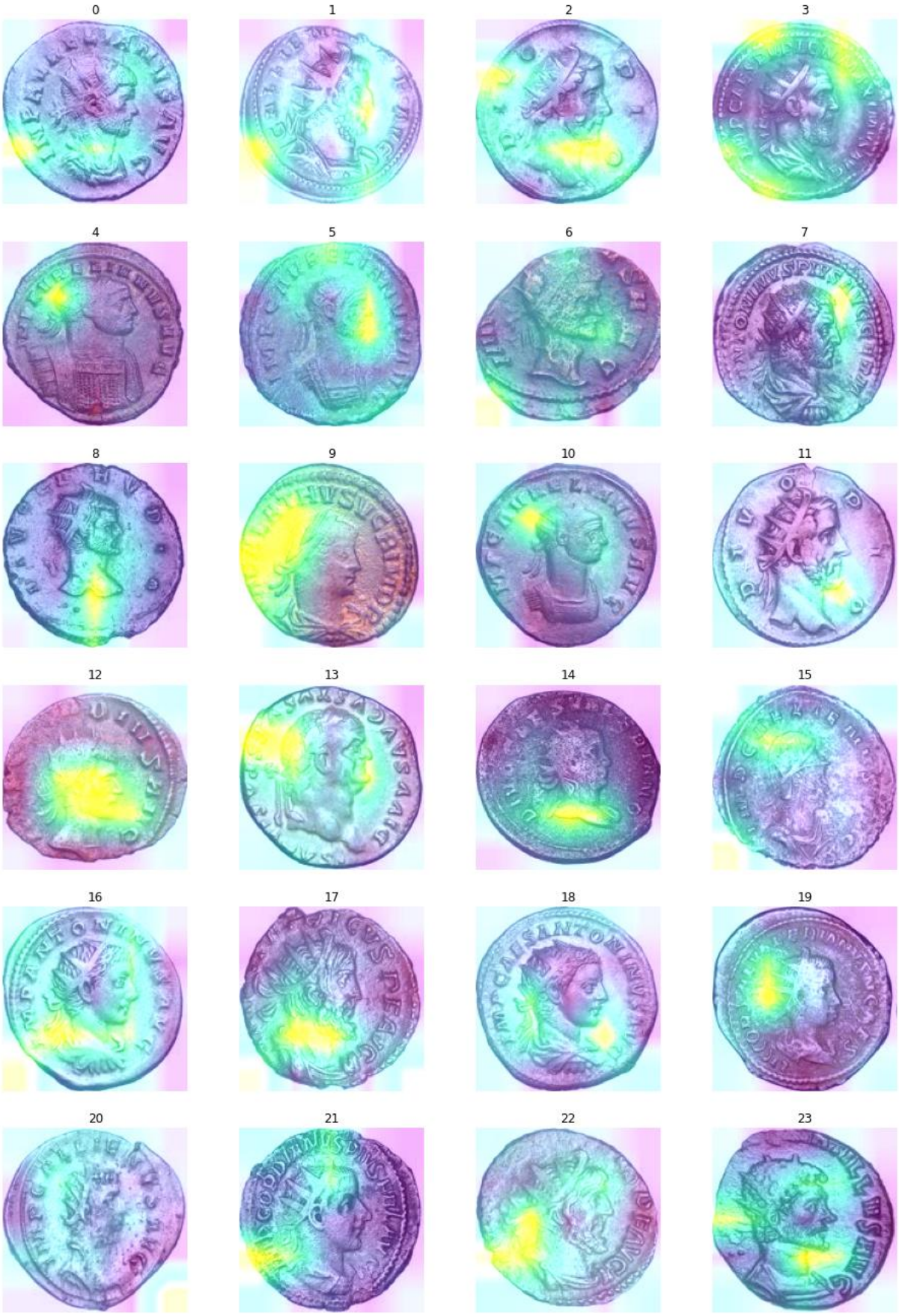


Figure 49: Grad-CAM of mislabelled antoniniani - Model 4 - Test 2 (excerpt).

Figure 50: Grad-CAM of mislabelled denarii - Model 4 - Test 2.

APPENDIX V: MODEL 6 Grad-CAM VISUALISATIONS - TEST 1

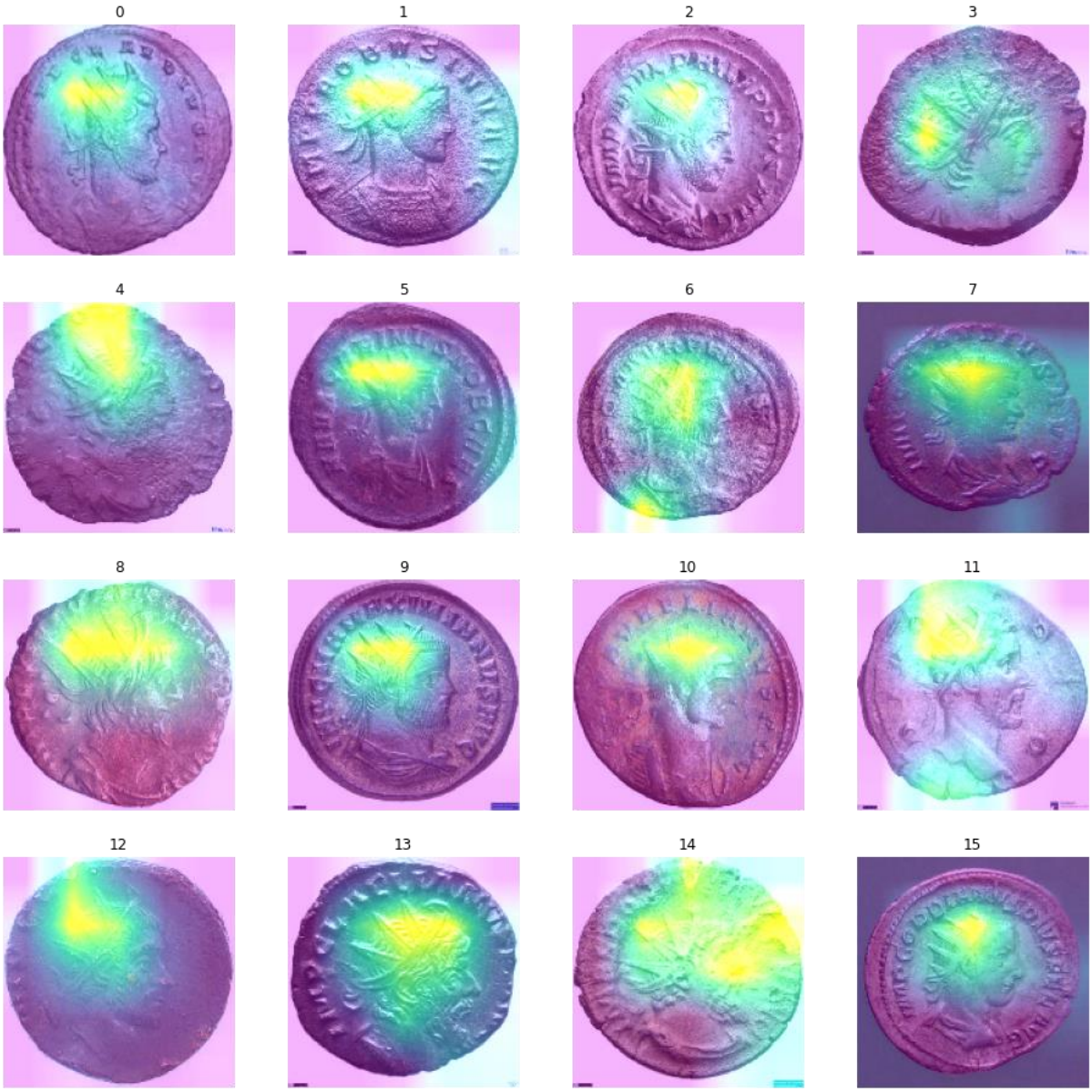


Figure 51: Grad-CAM of correctly labelled antoniniani - Model 6 - Test 1.

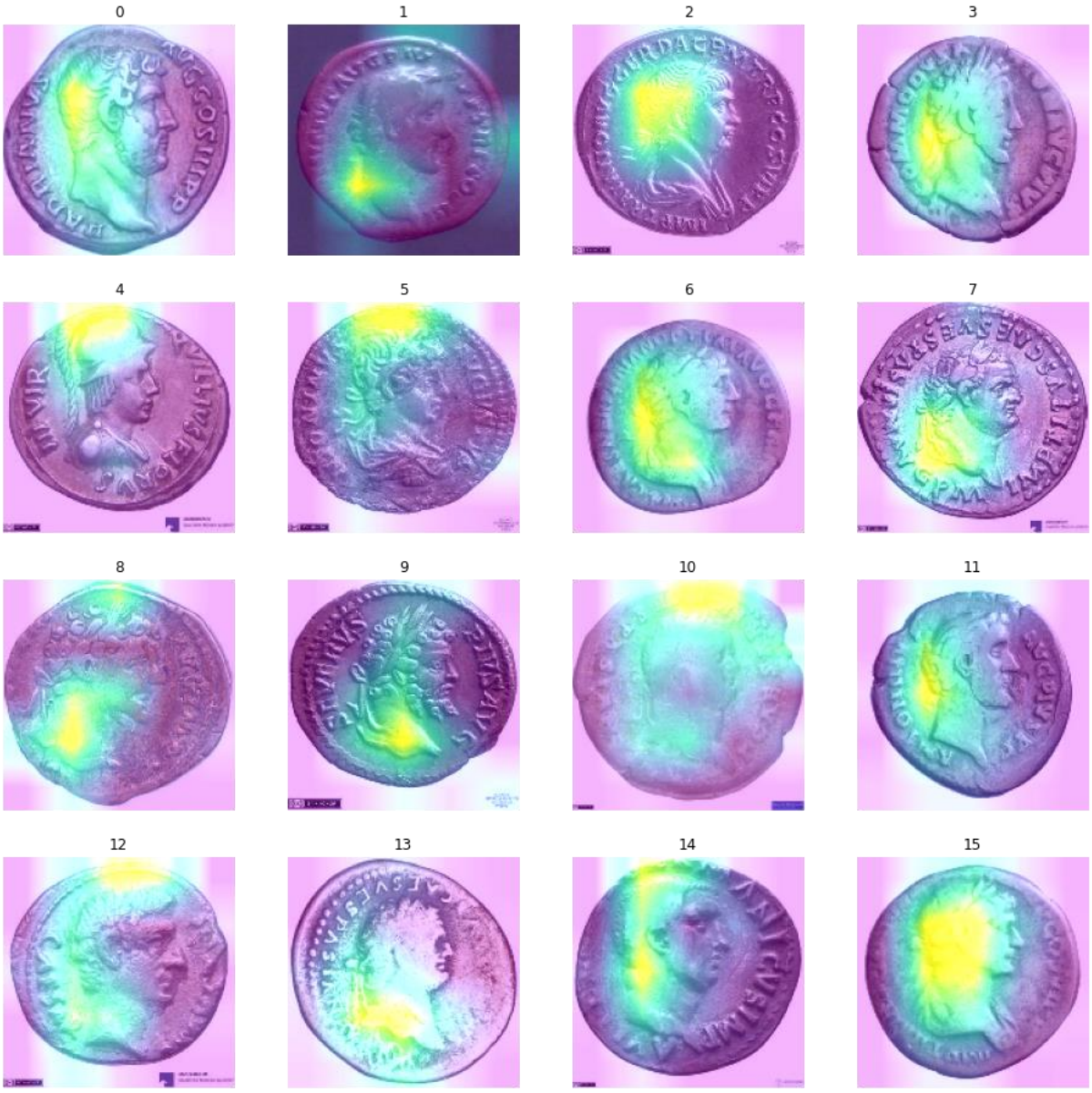


Figure 52: Grad-CAM of correctly labelled denarii - Model 6 - Test 1.

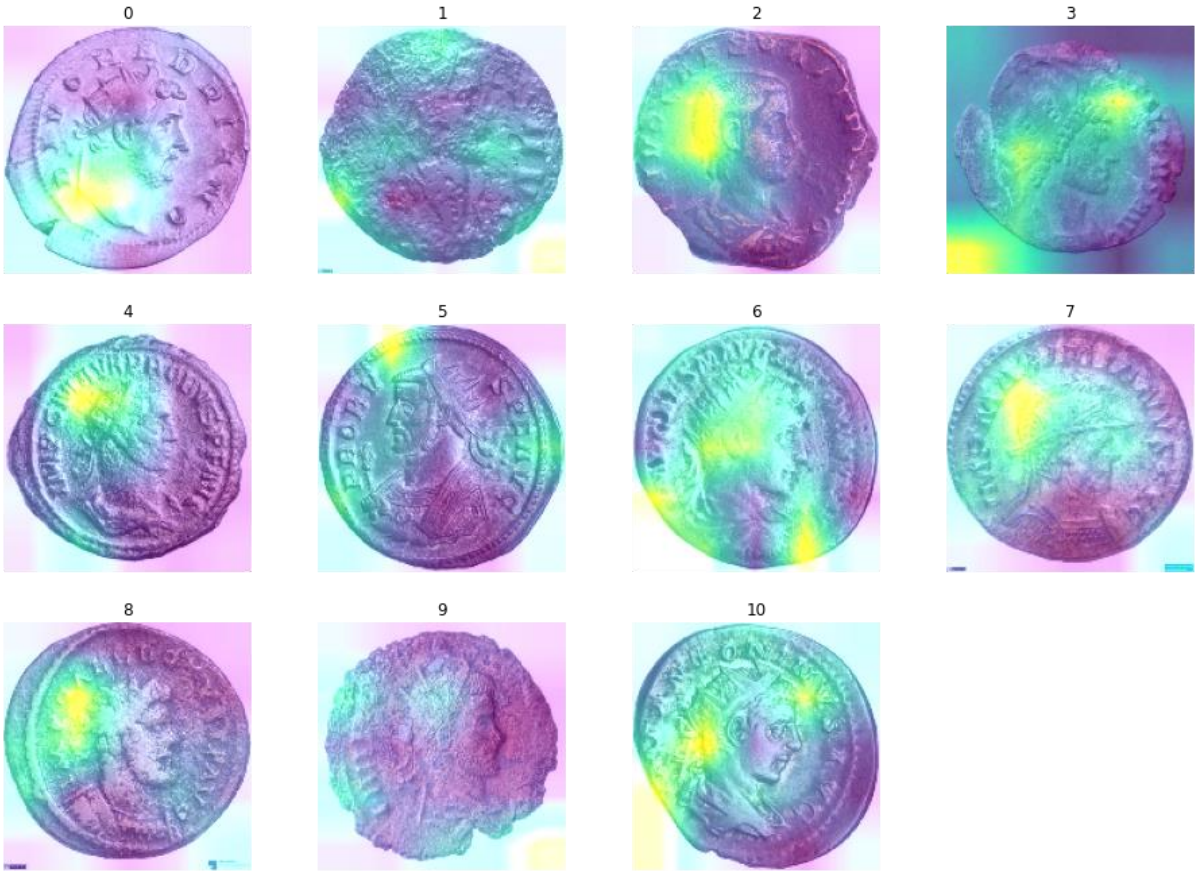


Figure 53: Grad-CAM of mislabelled antoniniani - Model 6 - Test 1.

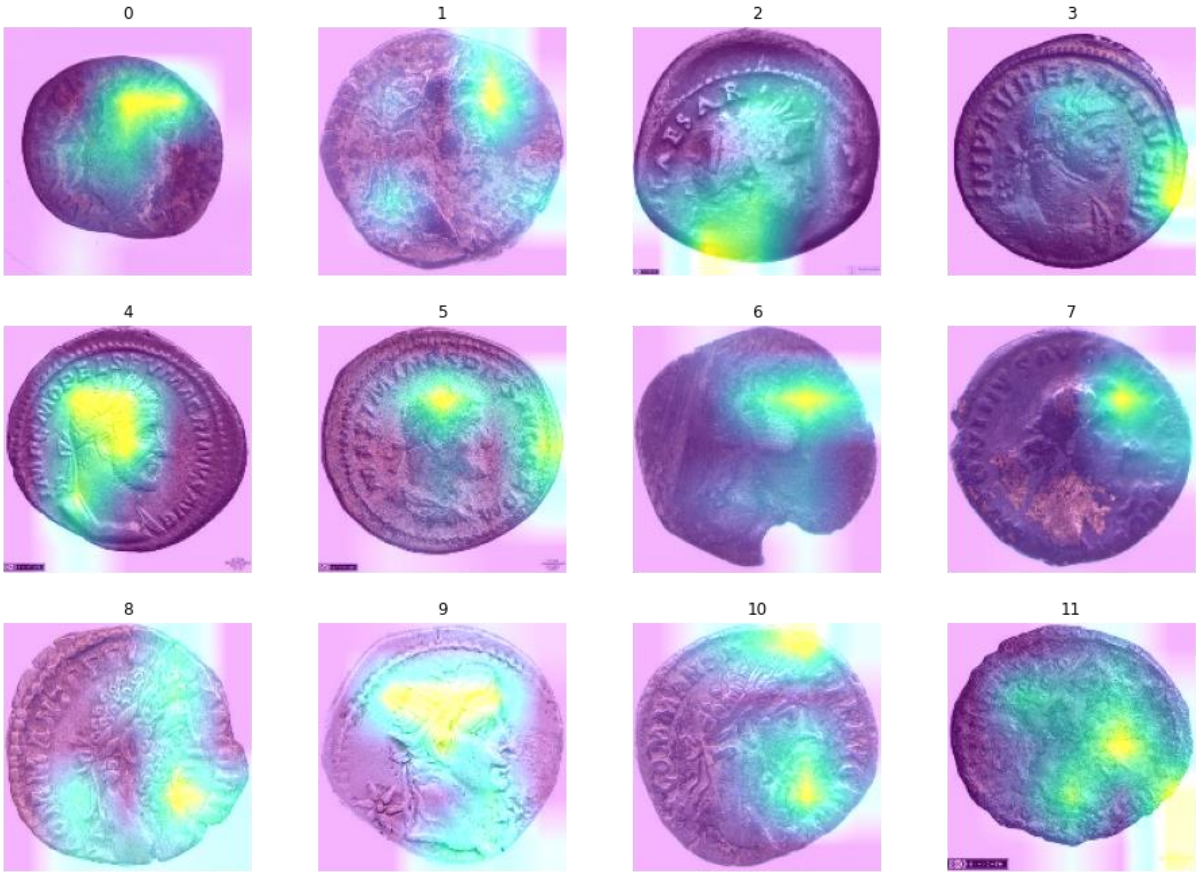


Figure 54: Grad-CAM of mislabelled denarii - Model 6 - Test 1.

APPENDIX VI: MODEL 6 Grad-CAM VISUALISATIONS - TEST 2

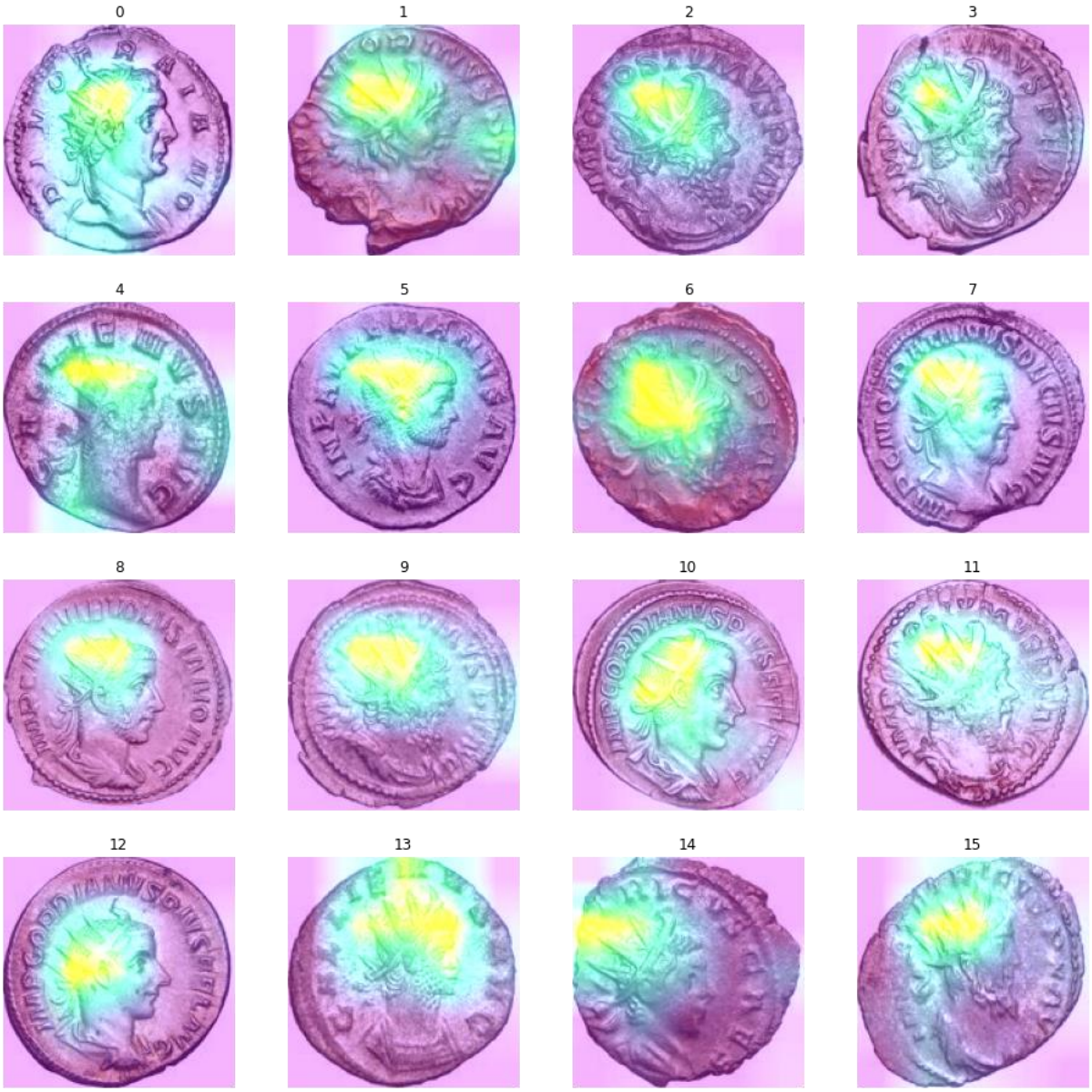


Figure 55: Grad-CAM of correctly labelled antoniniani - Model 6 - Test 2.

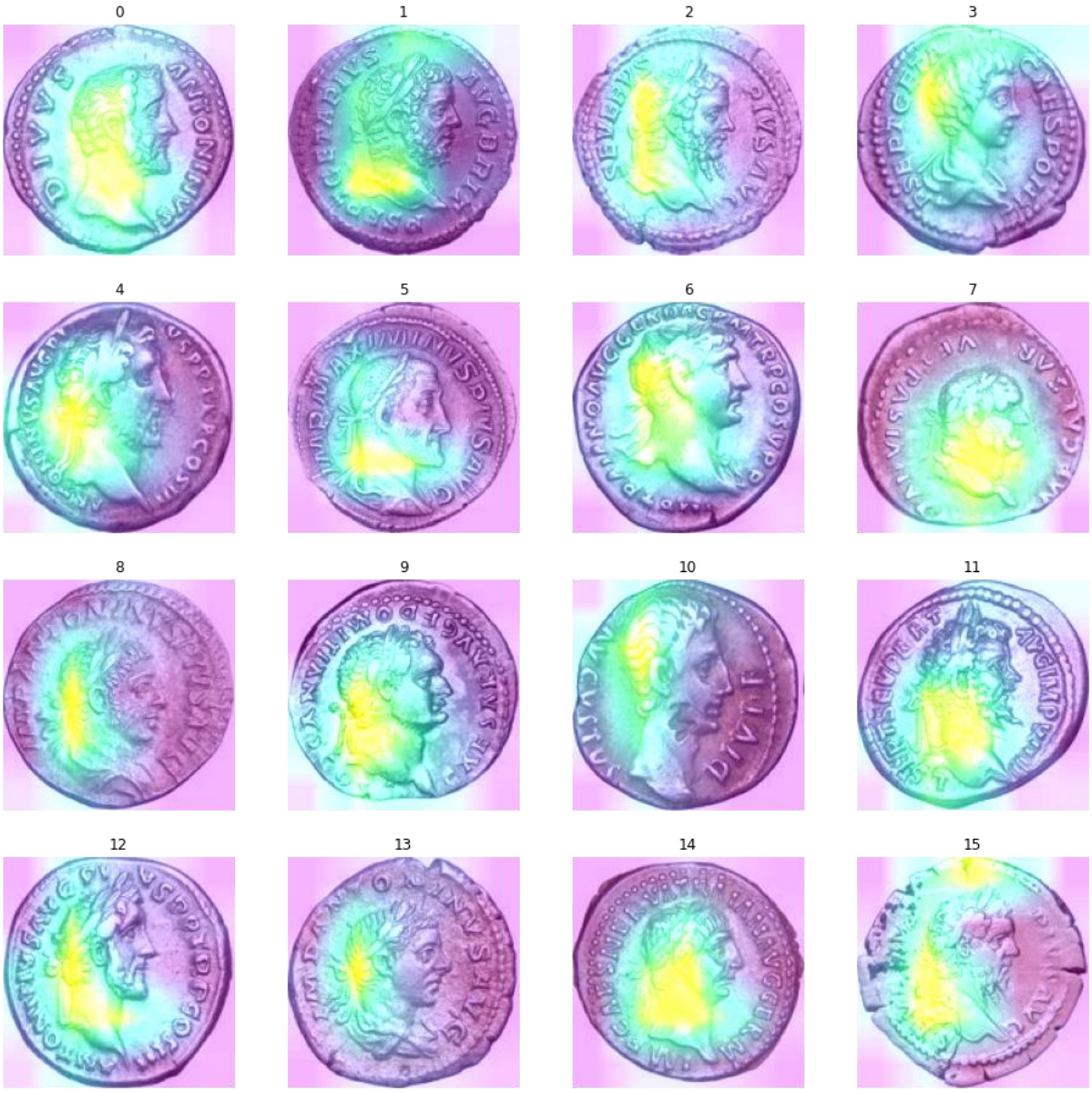


Figure 56: Grad-CAM of correctly labelled denarii - Model 6 - Test 2.

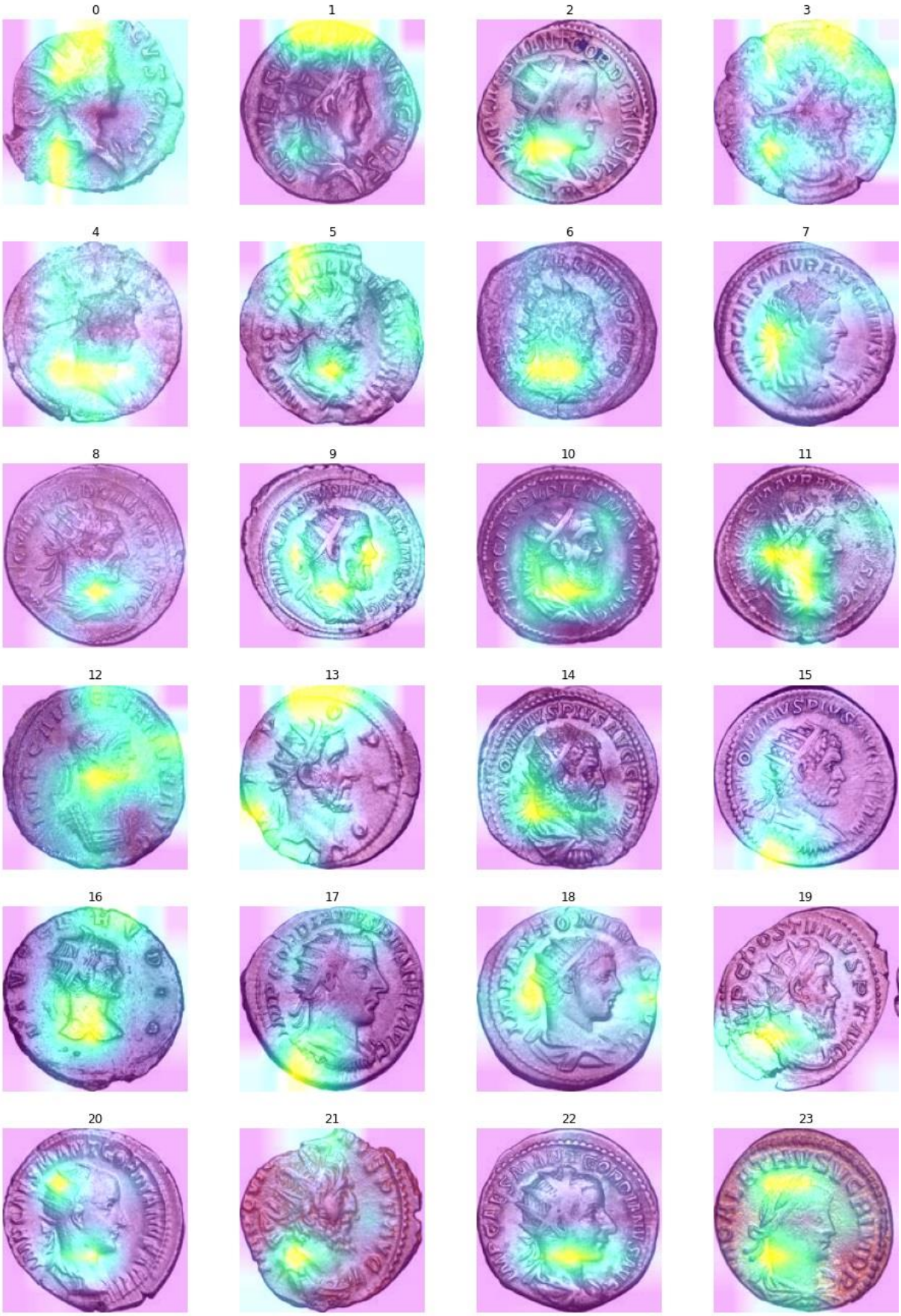


Figure 57: Grad-CAM of mislabelled antoniniani - Model 6 - Test 2 (excerpt).

Figure 58: Grad-CAM of mislabelled denarii - Model 6 - Test 2.