SCHOOL OF ENGINEERING

AND BUILT ENVIRONMENT

Image Processing & Machine Vision

Course Work 1

Author:	Luke Byrne
Matriculation Number:	S1823325
Program Code:	P02866
Program Title:	Electronic and Electrical Engineering
Module Code:	MMH623545-20-B
Module Title:	Image Processing and Machine Vision
Date of Submission:	15/03/2021
Word Count:	1,593

TABLE OF CONTENTS

List of Figures and Tables 1
1 Introduction 2
1.1 Aims and Objectives 2
2 Program Description 2
2.1 Overview 2
2.2 Pre-processing 3
2.3 Binaries 4
2.4 Segmentation 4
2.5 Non-coin Removal 5
2.6 Detecting 50 Pences 5
2.7 Coin classification 6
3 Testing and Results 7
3.1 Sample Set Results 7
3.2 Extra Image Set Results 7
4 Reflection and Conclusions 8
APPENDIX 11
Pre-processing Flow chart 11
Classification Flow Chart 12
Extra Image Set 13

LIST OF TABLES AND FIGURES

Table 1: Sample Set Results 7
Table 2: Extra Image Set Results 8

Figure 1: Program Flow 3
Figure 2: Coins3.jpg Pre-Processing Results 3
Figure 3: Coins3.jpg Pyramid Means Shifted Image Histogram 4
Figure 4: Coins3.jpg Binaries 4
Figure 5: Coins3.jpg Segmentation outputs 5
Figure 6: coins3.jpg Coins Found 5
Figure 7: coins3.jpg Straight Edges detected (highlighted in green) 6
Figure 8: coins3.jpg Values Identified 6
Figure 9: pre-Processing Flow Chart 11
Figure 10: Classification Flow chart 12
Figure 11: new1 Image 13
Figure 12: new1 histogram 13

Figure 13: New1 Binaries 13
Figure 14: New2 Image 14
Figure 15: New2 Histogram 14
Figure 16: New2 Binaries 14
Figure 17: New3 Image 15
Figure 18: New3 Histogram 15
Figure 19: New3 Binaries 15
Figure 20: New4 Image 16
Figure 21: New4 Histogram 16
Figure 22: New4 Binaries 16
Figure 23: New5 Image 17
Figure 24: New5 Histogram 17
Figure 25: New5 Binaries 17

1 INTRODUCTION

Counting coins manually is a cumbersome process, prone to error. As such, several approaches have been developed to count coins automatically. These typically either use mechanical action [1] or image processing to perform classification [2]. Machine vision approaches typically involve the use of machine learning techniques [3] [4]. However, the purpose of this project was to develop a program that would identify and classify coins using traditional image processing techniques such as thresholding, morphological operations, and segmentation. The program developed during this project uses bi-modal thresholding and watershed segmentation for coin detection, and a series of logical if statements assessing relative coin size and colour to perform classification.

1.1 AIMS AND OBJECTIVES

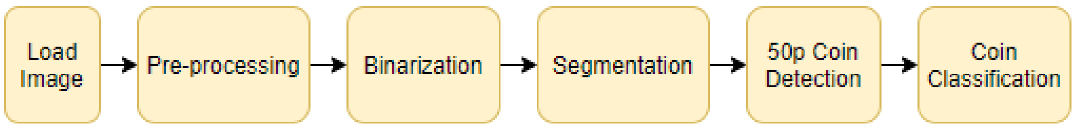
- Produce a python program that can identify and classify UK coins
- Write the program on Google Collab, using OpenCV and other free libraries
- Do not use machine learning
- Assess functionality using provided dataset and other images

2 PROGRAM DESCRIPTION

2.1 OVERVIEW

First, the program takes in images from the user. These images are pre-processed to make binarization easier. Binary masks are created via thresholding of different image channels. These binaries are used to perform watershed segmentation to identify objects. The objects identified are then kept or discarded based on their shape, reasoning that UK coins are almost perfectly circular and do not have holes in them. The program then tries to find 50 pences within the remaining coins via line detection. All other coins are classified based on colour, and their size relative to the 50 pences. This process is outlined in figure 1.

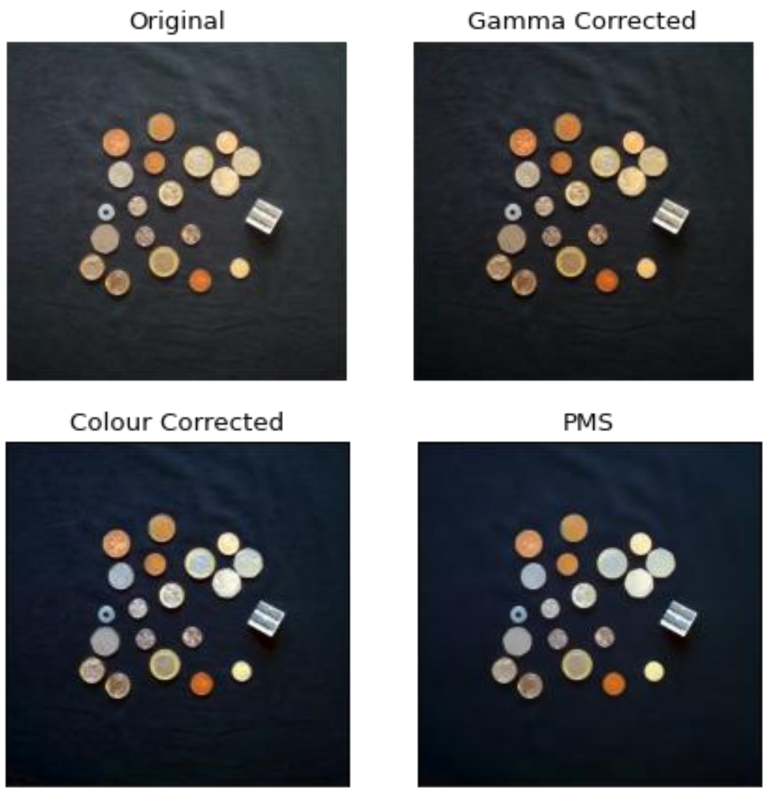
FIGURE 1: PROGRAM FLOW



2.2 PRE-PROCESSING

The aim of pre-processing is to increase the accuracy of later stages. As such, images are first resized to ensure that image size does not affect performance in images outside the sample set. Then gamma correction [5] is performed to skew the image histogram towards the extremes, making binarization easier. To improve colour identification, colour correction is performed using the same method as in Adobe Photoshop's auto levelling procedure [2] [3]. This process tends to brighten the background and so the value channel of the colour corrected image is replaced with that from the gamma-corrected image. Pyramid means shifting [8] is then performed to reduce detail. A complete flow chart for the pre-processing stage may be seen in figure 9 of the Appendix. The outcomes of pre-processing for coins3.jp may be seen in Figure 1.

FIGURE 2: COINS3.JPG PRE-PROCESSING RESULTS



2.3 BINARIES

Thresholding is used to produce binaries, for use in segmentation and coin classification. For segmentation, Otsu thresholding [9] is performed. Otsu thresholding is a technique for adaptive bimodal thresholding, appropriate in images that have histograms whose pixel intensities cluster around the extreme ends of the scale. As may be seen in figure 3, this is the case for images in the sample set. To aid in classification, two binary masks are also produced using hard-coded thresholds in the HSV colour space, one for copper and one for gold. Binary noise reduction is performed using morphological transformations [10]. The outputs of the Binarization stage for coins3.jpg may be seen in figures 3 and 4.

FIGURE 3: COINS3.JPG PYRAMID MEANS SHIFTED IMAGE HISTOGRAM

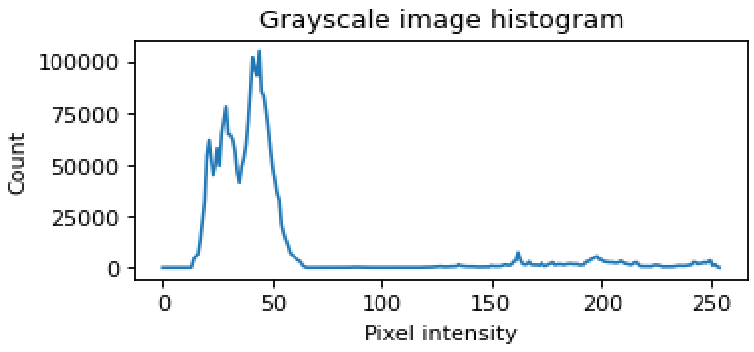
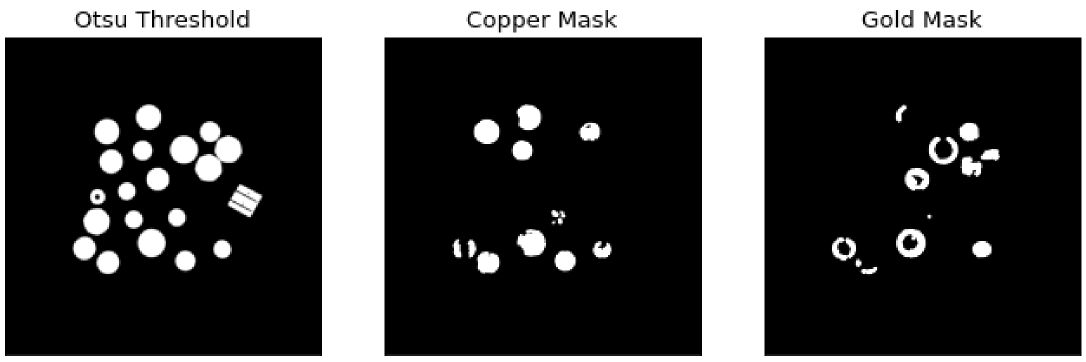


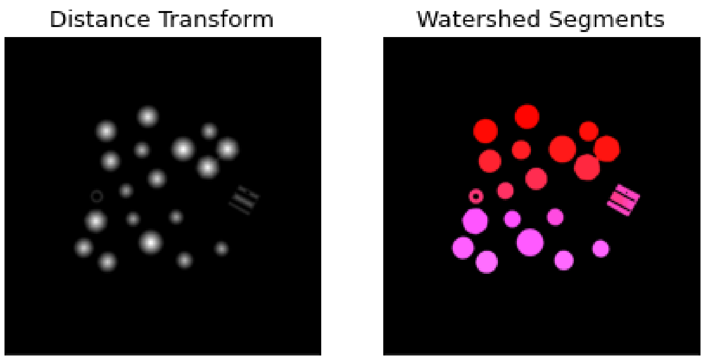
FIGURE 4: COINS3.JPG BINARIES



2.4 SEGMENTATION

Watershed segmentation [11] is then performed using the output of the Otsu thresholding. To do so, a distance transform is used to determine the distance of any 255 value pixel to the nearest 0 value pixel. Local peaks in the distance map are then merged, producing one peak per object. These peaks are used as the seed for a watershed algorithm, producing approximately one labelled segment per object. A visualisation of the distance map and watershed segments for coins3.jpg may be seen in figure 5.

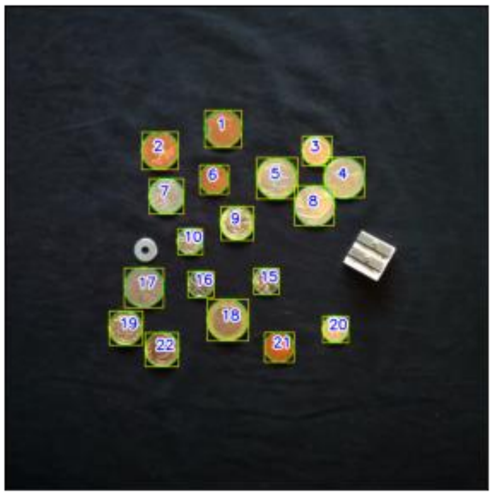
FIGURE 5: COINS3.JPG SEGMENTATION OUTPUTS



2.5 NON-COIN REMOVAL

At this stage, labels have been produced for each object highlighted during thresholding. However, non-coin objects remain. To remove these, a minimum enclosing circle [12] is computed for each label. The area of this circle is calculated. Then the actual area of the labelled segment is calculated. As coins are very round and do not have holes in them, only labels whose labelled area is within 80% of a perfect circle are retained. Bounding boxes are also calculated for each labelled object at this stage.

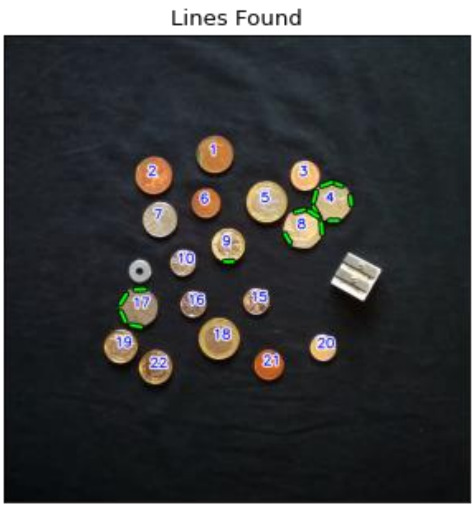
FIGURE 6: COINS3.JPG COINS FOUND



2.6 DETECTING 50 PENCES

Once non-coins have been removed, 50 pences are identified by their straight edges. First, Canny edge detection is performed on all labelled areas, followed by closing to ensure edge lines are contiguous [13]. A Hough Line transform [14] is then performed to detect straight lines in the edges. Coins that have at least 3 lines meeting a hard-coded minimum length are considered to be 50 pences. The outcome of this line detection for coins3.jpg may be seen in figure 6.

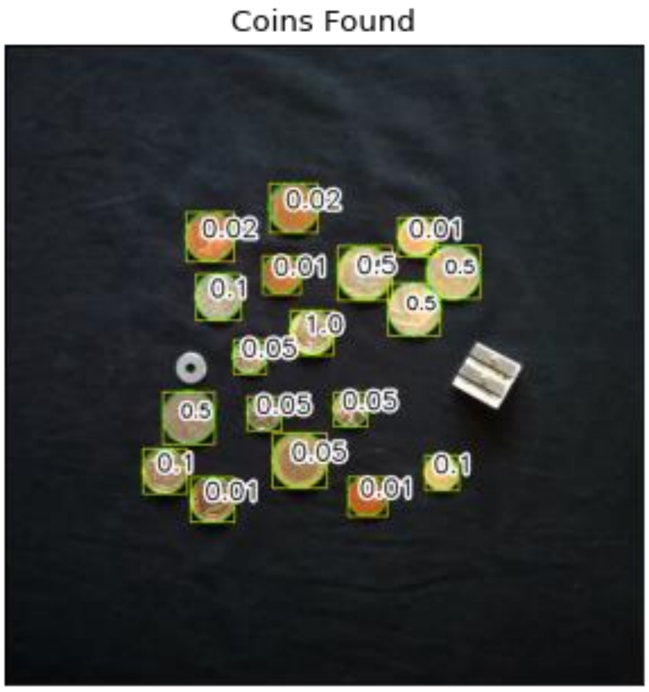
FIGURE 7: COINS3.JPG STRAIGHT EDGES DETECTED (HIGHLIGHTED IN GREEN)



2.7 COIN CLASSIFICATION

Coin classification is performed based on coin colour and size relative to known 50 pences. To do this, coin areas in the coin in the copper and gold masks are calculated. These areas are then compared to the area of a circle of best fit for the labelled region. If a coin has a significant relative copper area then it is considered a possible copper, and its size checked to see if it is a one or two pence. If a coin has a significant relative gold area then it is considered a possible pound, and its size checked to see if it is a one or two-pound. If a coin does not match these criteria then its size is checked to see if it is a 5, ten, twenty, or fifty pence. A complete flow chart for classification may be seen in figure 10 in the appendix.

FIGURE 8: COINS3.JPG VALUES IDENTIFIED



3 TESTING AND RESULTS

The program was tested on every picture in the sample set, and several additional images from outside the sample set. Performance was assessed in terms of the number of coins identified, the number of non-coins mistakenly included, and the number of coins values correctly identified.

3.1 SAMPLE SET RESULTS

TABLE 1: SAMPLE SET RESULTS

Image name	Number of coins	Number identified	Number of non-coins	Detected (%)	Number correct	Accuracy (%)
coins1	30	28	0	93.33	17	56.60
coins2	30	23	0	76.66	10	33.30
coins3	18	18	0	100.00	13	72.22
coins4	8	7	0	87.50	6	75.00
coins5	12	12	0	100.00	7	58.33
coins6	8	8	0	100.00	5	62.50
coins7	10	10	0	100.00	8	80.00
coins8	9	9	0	100.00	8	88.80
coins9	10	10	0	100.00	7	70.00
coins10	1	1	0	100.00	1	100.00

Table 1 shows the results of testing for images in the sample set. Detection was generally good, with the average across all images being 95.75%. A different thresholding algorithm than Otsu may have produced better results, as some coins were missed in the Otsu binaries. Other researchers have noted that the Kullback-Leiber Entropy thresholding method is better able to handle changes in lighting conditions [15]. No non-coins were incorrectly detected in any image, indicating that the non-coin removal process is effective.

Accuracy varied significantly between images (range 33.30% to 100%) with the average being 69.68%. From this, it is clear that colour and size descriptors are not specific enough to produce reliable results. Other descriptors using scale-invariant features for text and surface pattern recognition could likely be included to increase accuracy, as this method has been used by others in the past [2].

Coins2 saw both the lowest percentage of coin detection (76.66%) and accuracy (33.30%). This is likely due to the high amount of motion blur in the image. This causes the shape of coins in the image to become elongated along the axis of motion. Methods that specifically reduce motion blur would need to be implemented to improve detection and accuracy in this image.

3.2 EXTRA IMAGE SET RESULTS

A series of five extra images were included in testing to assess the robustness of program performance to different coloured and non-uniform backgrounds.

Backgrounds in the new images were black, wood, grey, red, and green. Please see the extra image set in the appendix to see the images included, as well as their resulting grayscale histograms and binaries. These results are shown in Table 2.

TABLE 2: EXTRA IMAGE SET RESULTS

Image name	Background type	# of coins	# identified	# of non-coins	Detected (%)	# correct	Accuracy (%)
new1	Black	16	13	1	21.31	1	6.25
new2	Wood	14	0	0	0.00	0	0.00
new3	Gray	14	7	0	50.00	0	0.00
new4	Red	14	0	0	0.00	0	0.00
new5	Green	15	0	0	0.00	0	0.00

Only the black and grey background images saw any coins identified. Consideration of the image histograms, available in the appendix, shows that only the black and grey images have histograms that cluster around the extremes. This makes the bimodal grayscale thresholding used in this program inappropriate for the wood, red and green background images.

However, even in the black and grey background images, the accuracy was not above 6.25%. This is because the 50 pence detection procedure broke down, and all coins were considered 50 pences. Although a resizing procedure is performed during the pre-processing, the minimum line length specified during the 50 pence detection phase is dependent on the distance between coins and the camera when the image is taken. This distance was not controlled for when taking the new images, and so accuracy suffered.

4 REFLECTION AND CONCLUSIONS

The program developed during this project had an average rate of coin detection of 95.75% for the sample image set, with no non-coins mistakenly included in the output. However, the average accuracy of coin classification was only 69.68%. Other methods for classification than logical if statements regarding coin descriptors may produce better results, such as k-means clustering [16], fuzzy logic classification [17], or use of scale and rotation invariant feature transforms [2]. Similarly, the inclusion of other descriptors beyond size and colour, such as text and pattern recognition, for each coin would likely increase accuracy [2]. A different approach to thresholding would likely result in better performance in images outside the sample set. Either by using a different thresholding algorithm [15] or by using a modified algorithm that directly samples the background [16].

Overall, this project served as a good introduction to the python programming language and introduced many interesting concepts in image processing. The relative complexity of classification was made clear, as was the necessity for rich object descriptors in object recognition.

5 REFERENCES

[1] Q. L. D. Li Ai Pan, “Study on Automatic Sorting and Counting Machine for Coin,” *Applied Mechanics and Materials,* Vols. 427-429, 2013.

[2] V. C. Kēyūra Dīnēshchandra Jōshī, “Real Time Recognition and Counting of Indian Currency Coins using Machine Vision: A preliminary Analysis,” in *The Canadian Society for Mechanical Engineering International Congress 2016*, 2016.

[3] R. Bremananth, “A new approach to coin recognition using neural pattern analysis,” in *IEE INDICON*, 2005.

[4] V. C. M., P. Vivekanadan and K. R. Kashwan., “Indian coin recognition and sum counting system of image data mining using artificial neural networks,” *International Journal of Advanced Science and Technology,* vol. 31, pp. 67-80, 2011.

[5] OpenCV, “Changing the contrast and brightness of an image!,” [Online]. Available: https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html. [Accessed 28 March 2021].

[6] “Illuminant Estimation: Simplest Color Balance,” [Online]. Available: https://web.stanford.edu/~sujason/ColorBalancing/simplestcb.html. [Accessed 26/03/2021 March 2020].

[7] D. Y. Kay, “Simple Color Balance Python Github,” [Online]. Available: https://gist.github.com/DavidYKay/9dad6c4ab0d8d7dbf3dc. [Accessed 26/03/2021 March 2020].

[8] OpenCV, “Meanshift and Camshift,” [Online]. Available: https://docs.opencv.org/master/d7/d00/tutorial_meanshift.html. [Accessed 28 March 2021].

[9] OpenCV, “Image Thresholding,” [Online]. Available: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html. [Accessed 28 March 2021].

[10] OpenCV, “Morphological Transformations,” [Online]. Available: https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html. [Accessed 28 Mrch 2021].

[11] OpenCV, "Image Segmentation with Distance Transform and Watershed Algorithm," [Online]. Available: https://docs.opencv.org/3.4/d2/dbd/tutorial_distance_transform.html. [Accessed 28 March 2021].

[12] OpenCV, “Creating Bounding Boxes and Circles for Contours,” [Online]. Available:

https://docs.opencv.org/3.4/da/d0c/tutorial_bounding_rects_circles.html. [Accessed 28 March 2021].

[13] OpenCV, “Canny Edge Detection,” [Online]. Available: https://docs.opencv.org/master/da/d22/tutorial_py_canny.html. [Accessed 28 March 2021].

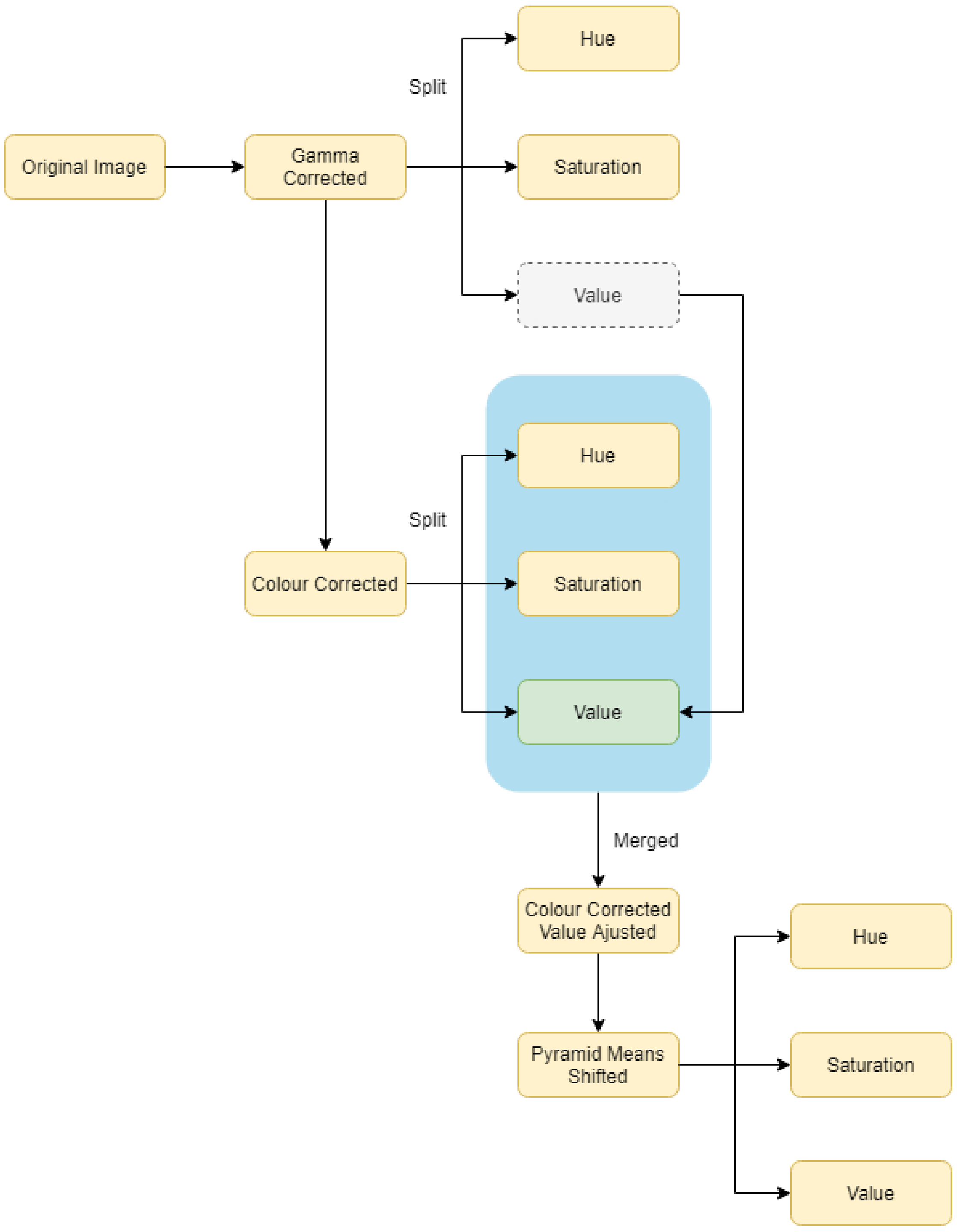
[14] OpenCV, “Hough Line Transform,” [Online]. Available: https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html. [Accessed 28 March 2021].

[15] S. D. N. D. Payel Roy, “Adaptive Thresholding: A comparative study, Goutami Dey, Sayan Chakraborty, Ruben Ray,” in *International Conference on Control, Instrumentation, Communication and Computational Technologies*, 2014.

APPENDIX

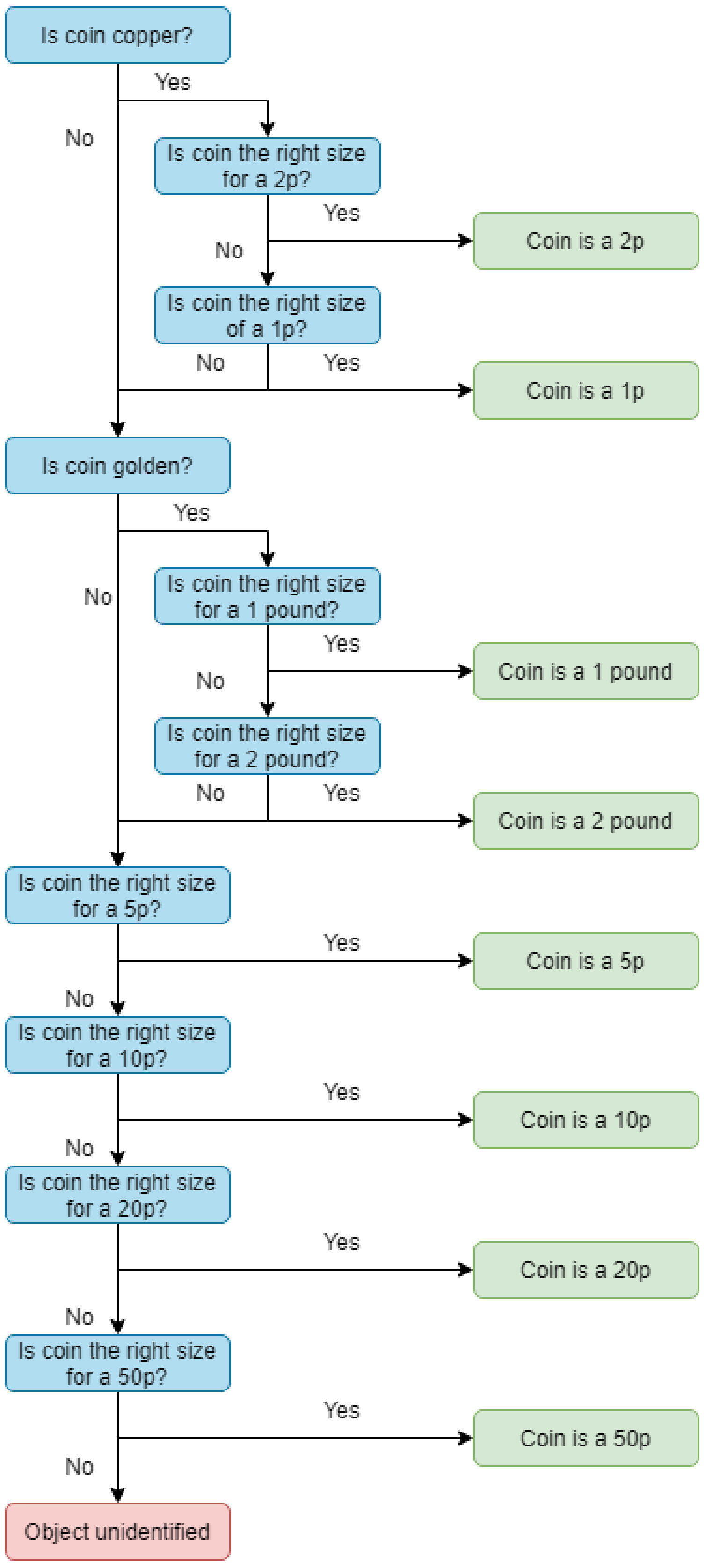
PRE-PROCESSING FLOW CHART

FIGURE 9: PRE-PROCESSING FLOW CHART



CLASSIFICATION FLOW CHART

FIGURE 10: CLASSIFICATION FLOW CHART


Is coin copper?
Yes
No
Is coin the right size for a 2p?
Yes
Coin is a 2p
No
Is coin the right size of a 1p?
No
Yes
Coin is a 1p
Is coin golden?
Yes
No
Is coin the right size for a 1 pound?
Yes
Coin is a 1 pound
No
Is coin the right size for a 2 pound?
No
Yes
Coin is a 2 pound
Is coin the right size for a 5p?
Yes
Coin is a 5p
No
Is coin the right size for a 10p?
Yes
Coin is a 10p
No
Is coin the right size for a 20p?
Yes
Coin is a 20p
No
Is coin the right size for a 50p?
Yes
Coin is a 50p
No
Object unidentified

EXTRA IMAGE SET

FIGURE 11: NEW1 IMAGE

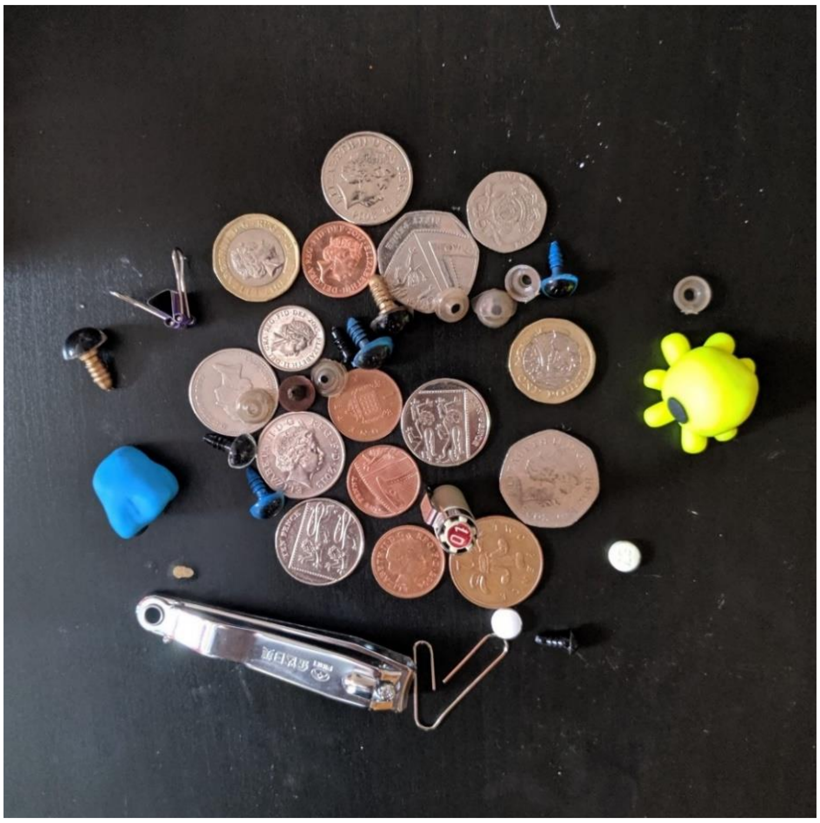


FIGURE 12: NEW1 HISTOGRAM

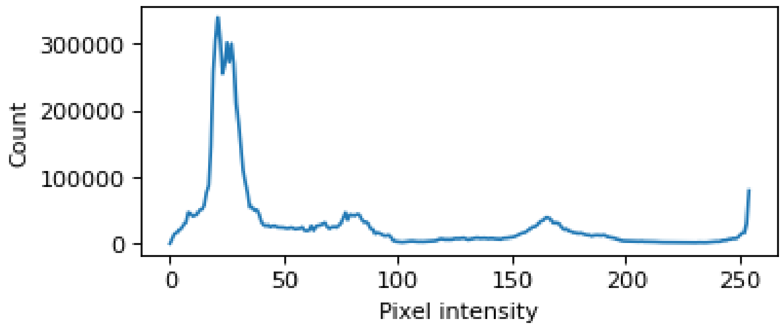


FIGURE 13: NEW1 BINARIES

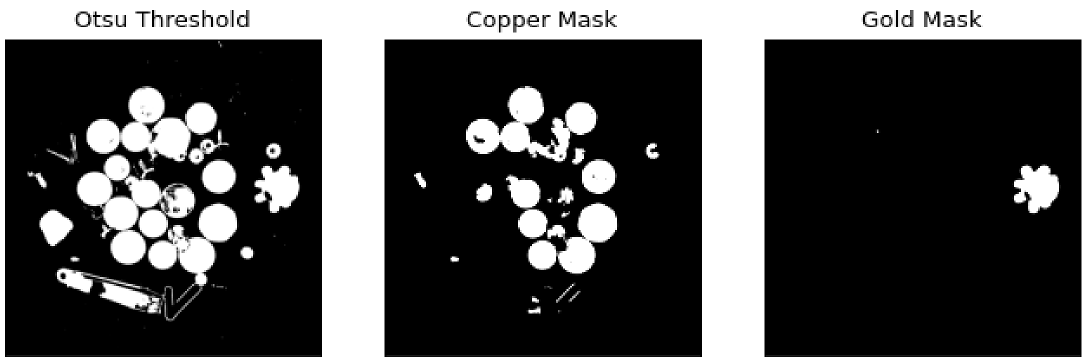


FIGURE 14: NEW2 IMAGE

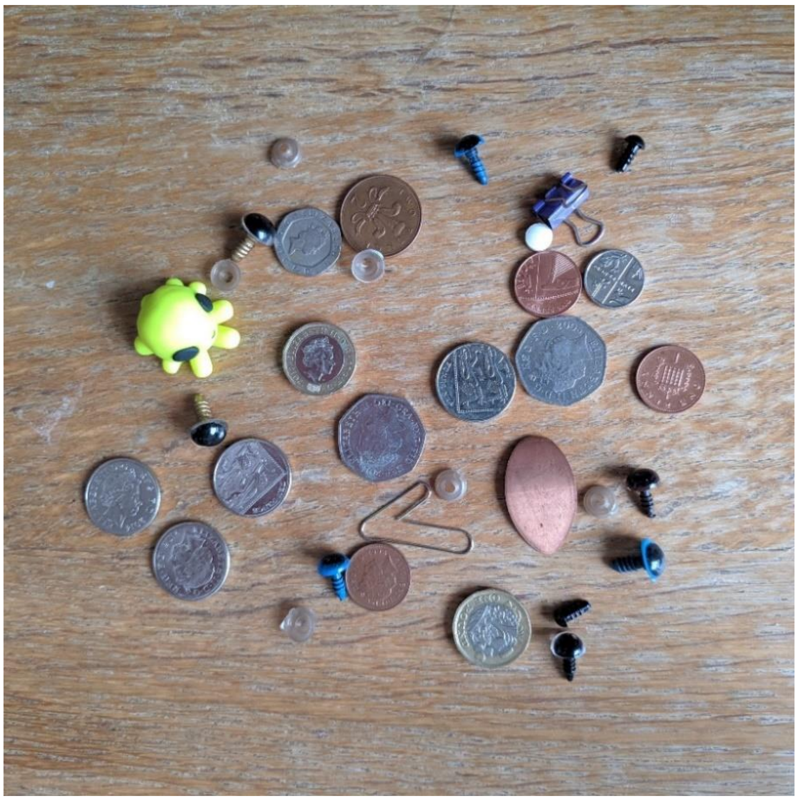


FIGURE 15: NEW2 HISTOGRAM

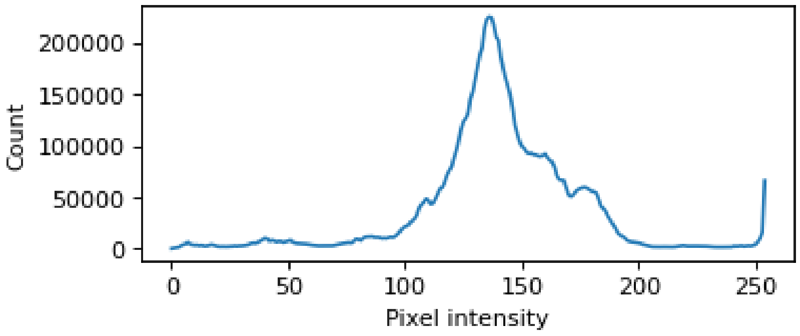


FIGURE 16: NEW2 BINARIES

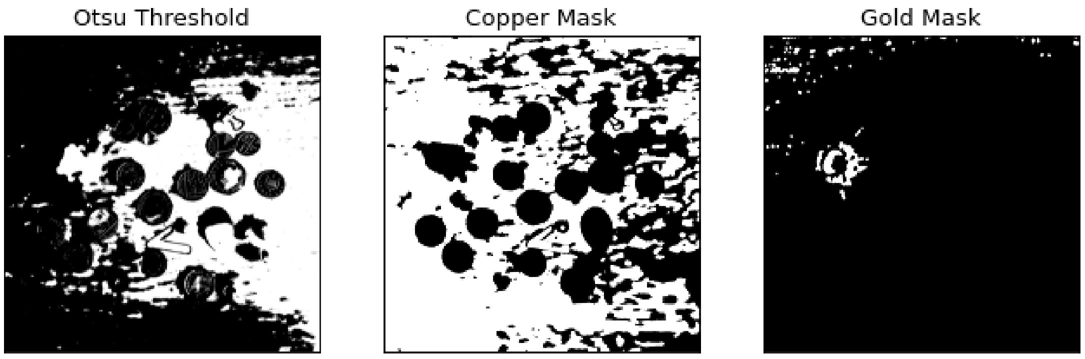


FIGURE 17: NEW3 IMAGE

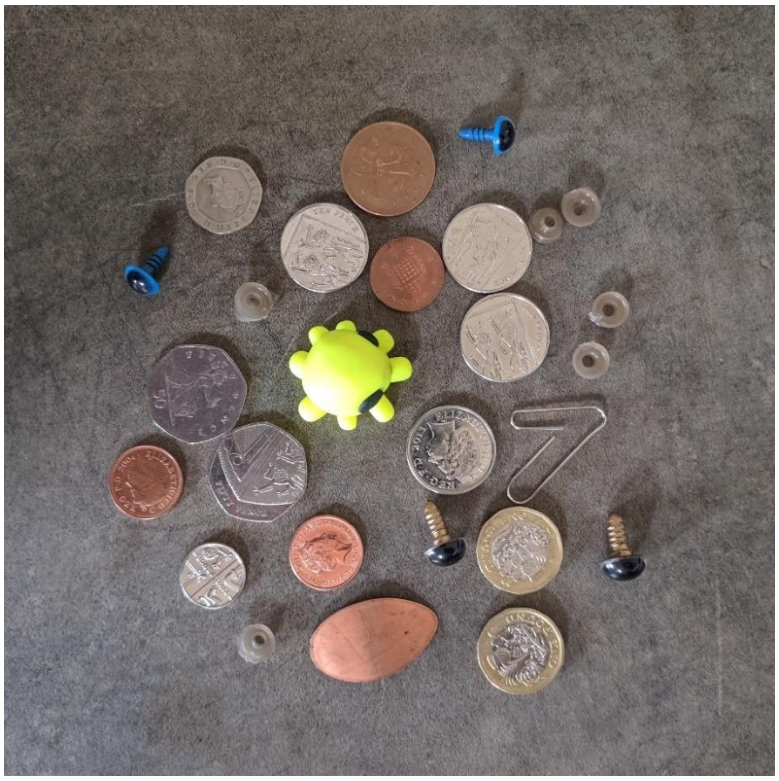


FIGURE 18: NEW3 HISTOGRAM

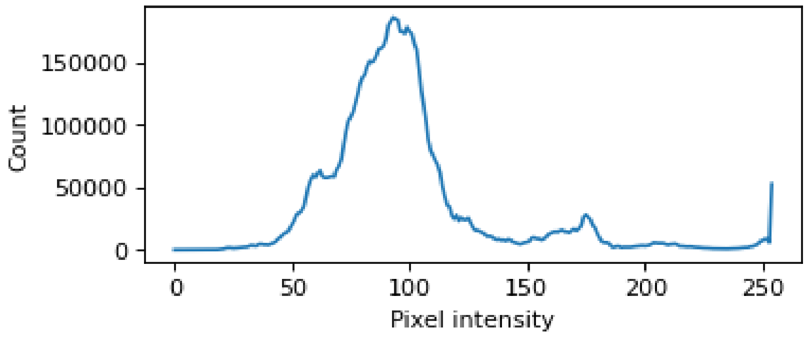


FIGURE 19: NEW3 BINARIES

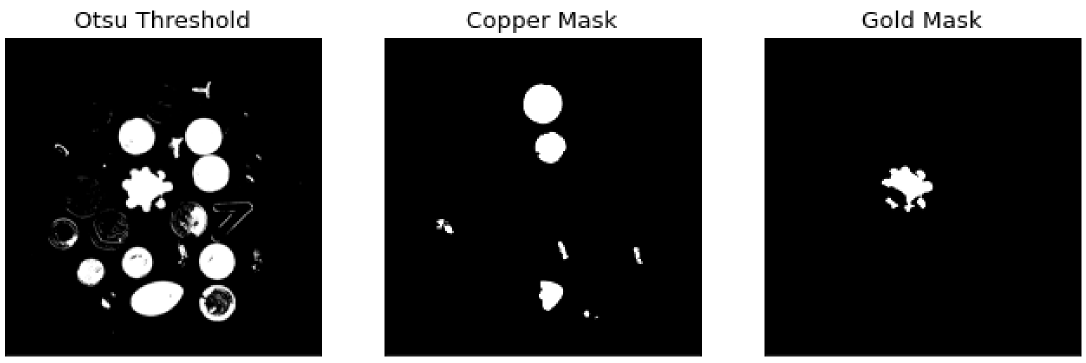


FIGURE 20: NEW4 IMAGE

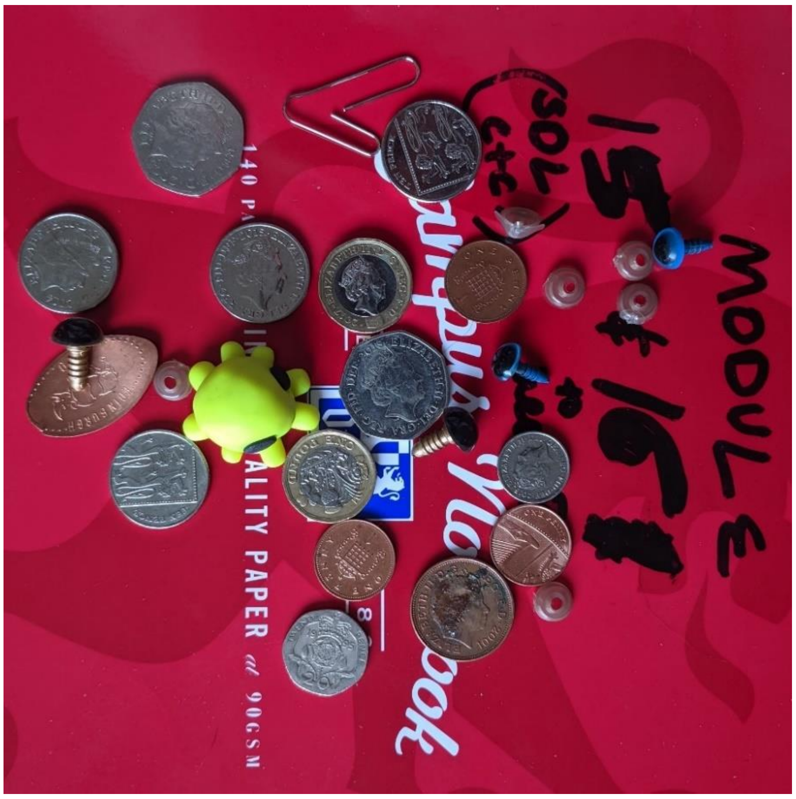


FIGURE 21: NEW4 HISTOGRAM

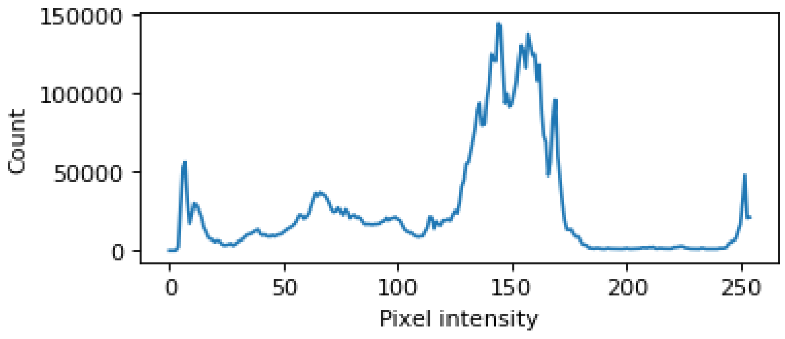


FIGURE 22: NEW4 BINARIES

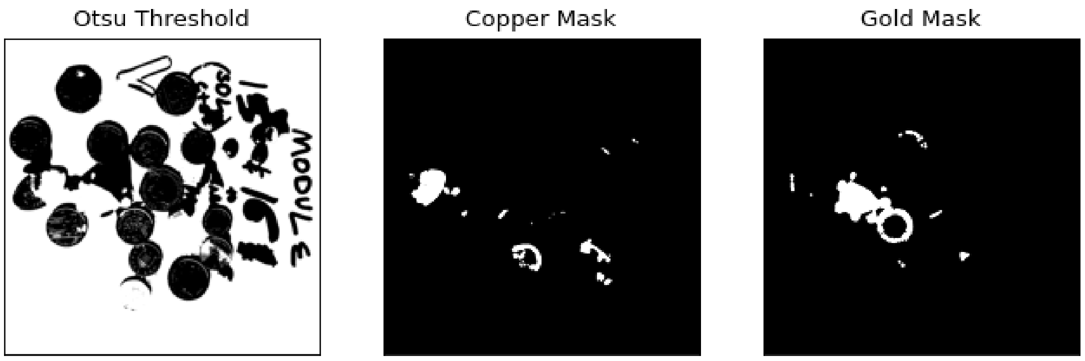


FIGURE 23: NEW5 IMAGE

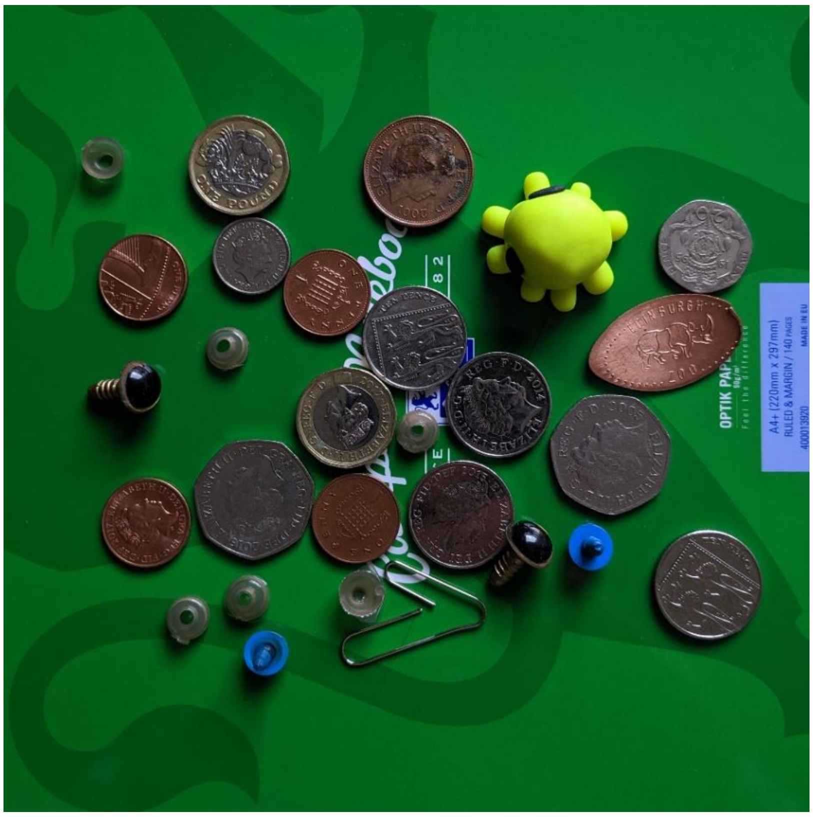


FIGURE 24: NEW5 HISTOGRAM

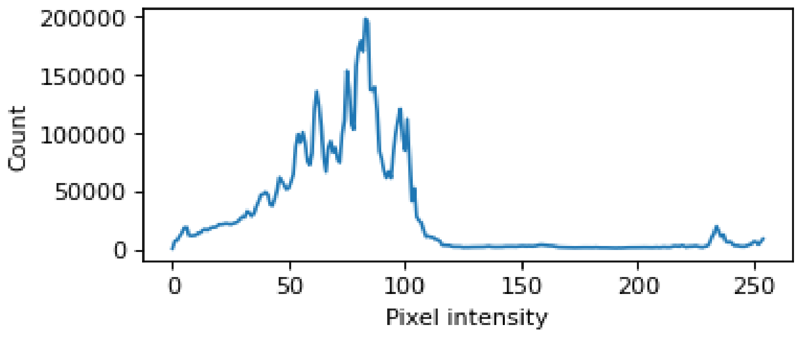
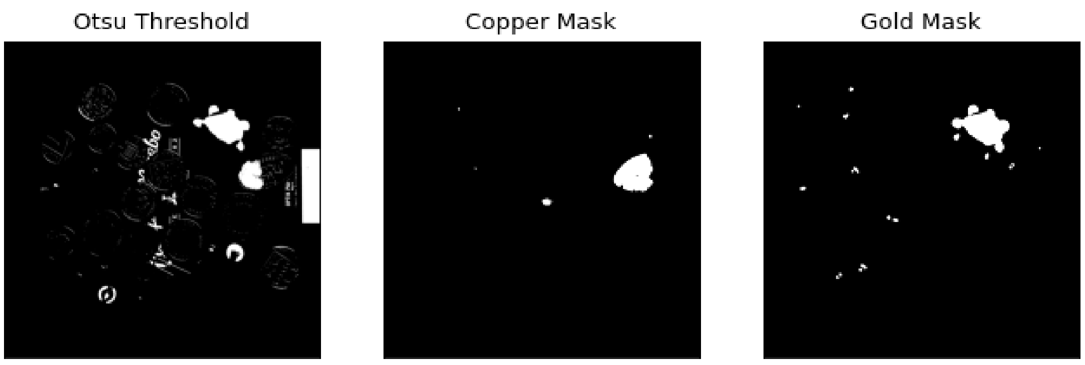


FIGURE 25: NEW5 BINARIES


Otsu Threshold
Copper Mask
Gold Mask