Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/14/20230795

Brazilian coin counter research report

Yian Wu[1,5], Xiran Lin[2,6,†], Xie Wu[3,7,†] and Han Cheng[4,8]

[1]The Study, Montreal, H3Y 1S4, Canada
[2]Beijing21st Century International School, Beijing, 100142, China
[3]Eberly college of science, The Pennsylvania State University, State College, 16803, United States
[4]Tsinglan School, Dongguan, 523429, China, Hannah

[5]yian060223@gmail.com
[6]Lin_xiran@hotmail.com
[7]wuxiewx1022@163.com
[8]Cheng_25@tsinglan.org
[†]Xiran Lin and Xie Wu contributed equally to this work and should be considered as co-second authors.

Abstract. Though coins are less used in the informational era, a significant amount of the population still uses physical coins for financial transactions. Also, when the country's government collects the coins for equivalent digital currency transactions, coin identification is still vital yet tedious. Moreover, the dated coins and the difference in illuminations. This novel presents a coin identification and validation model based on the Alex Net convolutional model. It identifies the value of a coin through the numbers, animals, and plants on the two sides. The model employs data enhancement, feature attention layers, max pooling, and residue groups. We collected 2826 images of Brazilian coins with reverse motifs, and the experimental accuracy of our model reached 0.97. The code part has shown here: https://github.com/Erik-Xie/-Brazilian-Coin-counter-Research-Report.git

Keywords: coin validation and identification, Alex Net convolutional network.

1. Introduction

Coins are small, circular discs of metal primarily used as a form of money for daily transactions. A coin's exchange value is determined and certified by marks or engravings on both sides. However, as coins wear down over time, their imprints become hard to distinguish by the eye, leading to difficulty in recognizing their values. Moreover, because coins are most often used for lower-valued units, manually counting, sorting, and calculating the total value of many coins is usually repetitive and tedious. Therefore, machine learning can significantly facilitate this task since computers can promptly classify numerous coins' values if trained with adequate image classification models. The primary purpose of this thesis is to develop an accurate coin detector to classify different types of Brazilian coins using convolutional neural networks (CNN). In this paper, the Literature Review is presented in Chapter II, Methodology in Chapter III, Exploratory Data Analysis in Chapter IV, Experimental Settings in Chapter

DOI: 10.54254/2755-2721/14/20230795

V, Evaluation Metrics and Results in Chapter VI, Experimental Results in Chapter VII, Discussions and Limitations in Chapter VIII, and Conclusion in Chapter IX.

2. Literature review

Numerous ways can be used to recognize coins, among which the most common is human eye recognition. However, human eye recognition is often highly time-consuming when faced with many coins. Several machines, such as parking toll devices, vending machines, and public telephones, classify coins by size and weight. Additionally, optical sensors have been invented to take two-dimensional images of coins. Then, these images are matched to the machine's existing templates, allowing rapid identification [1]. Machine learning methods have also been implemented to further improve recognition speed.

Pham, T.D. *et al.* [2] proposed a deep learning method to study the detection of fake banknotes. They constructed a two-stage classification model, which consisted of a first detector based on "you only look once, version 3" (YOLOv3), followed by a second convolutional neural network (CNN) classifier. Their experimental results demonstrate that this two-step method combining YOLO v3 and CNN showed higher accuracy than conventional detection models solely based on YOLO v3.

Park et al. [3] recognized that previous banknote and coin detection studies utilizing deep convolutional neural networks (CNN) showed reduced performance accuracy due to environment and background changes. To overcome these drawbacks, Park et al. [3] used faster region-based CNN, geometric constraints, and the residual network (ResNet) to create a three-stage model that detects banknotes and coins. Using the DKB v1 and the JOD open databases, their method proved to yield higher accuracy than the state-of-the-art methods, which include Faster R-CNN (7), MobileNet (4), YOLOv2 (8), YOLO v3 (6), and SURF (1-3&5) based detection methods with handcrafted features or deep features. However, despite the high accuracy for banknote detection, their proposed method produced false positive errors when detecting coins.

Xiang & Yan [4] studied the effectiveness of deep learning methods for identifying fast-moving coins. They built a model based on long and short-term memory (LSTM) combined with a convolutional neural network (CNN) to accurately recognize pictures of coins in motion. Their experimental results showed that LSTM combined with CNN effectively improved the accuracy of identifying fast-moving coins compared to their other model, which was solely based on Faster R-CNN.

Through the above studies on coin recognition in various situations, we can understand that CNN (Convolutional Neural Networks) (Figure 1), one of the representative algorithms of deep learning, is widely used in image segmentation and object recognition. We are very curious about its accuracy when processing images. In this article, we will use CNN to analyze the Brazilian Coins dataset [5] on Kaggle and analyze the accuracy and feasibility of this method through the visualization view.

3. Methodology – convolutional neural network

We apply the classic AlexNet convolutional neural network for our coin recognition algorithm. This model includes convolutional, max pooling, normalization, full-connection, and softmax layers. Every convolutional layer uses maximum pooling instead of mean pooling for downsampling a picture's detail so the data size is reduced without affecting the output. Instead of keeping the picture's overall feature, this helps identify the specific pattern on the coin and reduces overfitting.

$$W\_out\ = [\ (W - F + 2P)/S]\ +\ 1$$

$$H\_out\ = [(H - F + 2P)/S]\ +\ 1$$

$$D\_out\ =\ K$$

Convolution, where W is the input width, H is the input height, F is the filter size, P is the padding, and K is the number of filters.

$$r = 0,1,2,3,\cdots,p-1 \quad s = 0,1,2,3,\cdots,q-1$$

DOI: 10.54254/2755-2721/14/20230795

$$y_{ij} = max(x_{i \cdot t+r, j \cdot t+s})$$

$$i \leqslant (m-p)/t \;\; j \leqslant (n-q)/t$$

max pooling, where x is an m*n matrix, kernel size is p*q, t is the stride

Rectified linear unit (ReLU) is applied after every pooling layer as nonlinearities. It preserves the significant features of the previous layer's output as the input of the next layer, improving the model's training. It also prevents vanishing gradient problems and overfitting through constant output with positive input.

ReLU activation function: $f(x) = max(0, x)$

Furthermore, random dropout of specific neurons in the AlexNet convolutional network prevents overfitting. Moreover, the AlexNet convolutional network enhances data by randomly sampling a picture or reflection to magnify the data size, preventing overfitting.

All the classified patterns are assembled in the fully connected layers to output the predicted image result. These fully connected layers take the output from the last pooling layer to perform classification.

In this project, the architecture of our network is relatively simple, so we do not apply residual blocks to improve our network's efficiency, since it is not necessary for limited layers.

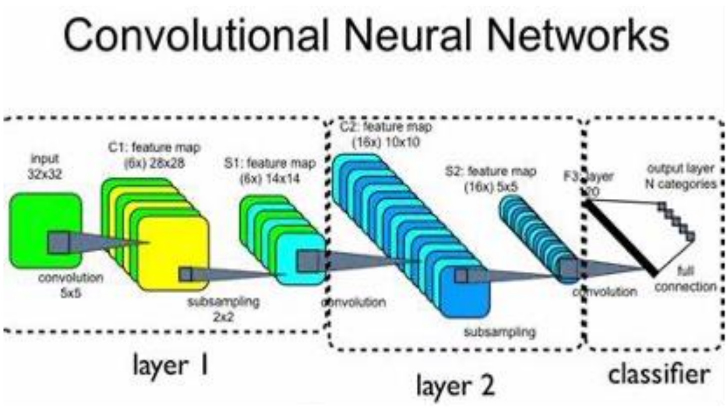

Figure 1. CNN flowchart.

4. Exploratory data analysis

4.1. Data acquisition and preprocessing

The experimental dataset used in this study is the Brazilian Coins dataset [5] on Kaggle.com, contributed by Luis Moneda.

This dataset includes two sub-datasets, one for classification problems and another for regression problems. Since this study aims to classify different types of coins, we used the sub-dataset for classification problems, which consists of 3059 JPG image files, each containing one single Brazilian coin. The images illustrate five types of coins: the 5 cents coin, the 10 cents coin, the 25 cents coin, the 50 cents coin, and the 100 cents coin. These five different classes make this experiment a multi-class classification task. Each image in the dataset shows a single coin on a plain background, with the coin facing upward. Slight variations can be noticed between the backgrounds and illuminations for each image, and each coin reflects light differently depending on their position.

Each file contains its value of money as its file name. We can see examples as below (Figure 2):

Figure 2. Samples of the data file.

The whole dataset is split into the training set and the test set. The training set consists of 2533 images, and the test set consists of 526 images, resulting in a train to test ratio of approximately 80:20.

The purpose of this dataset is to provide a resource for machine learning and computer vision applications that require images of Brazilian coins. This dataset can be used for object recognition, classification, and localization tasks.

4.2. Statistical analysis

Here are some descriptive statistics about the “Brazilian coins” dataset:

The dataset contains images of Brazilian coins from the following denominations: 5, 10, 25, and 50 centavos, as well as 1 real. The mean diameter of the coins in the dataset is approximately 24.3 millimeters, with a standard deviation of 4.5 millimeters. The mean weight of the coins in the dataset is approximately 5.6 grams, with a standard deviation of 4.0 grams. The most common coin denomination in the dataset is 1 real, followed by 50 centavos and 25 centavos. The earliest coins in the dataset were produced in 1901, and the most recent coins were produced in 2020. The dataset contains coins with different designs and images, such as portraits of historical figures, animals, and cultural symbols.

5. Experimental settings

We want to train a program to recognize coins using deep learning, and then the program can identify the denomination of the coins on its own. In deep learning, Convolutional Neural Networks (CNNs) have always been the most widely used method in image recognition. Therefore, we used this method to train the program's recognition ability.

We designed 5 hidden layers in the CNN, with parameters of 16, 32, 64, 128, and 256, respectively. After the convolution, we used the MLP method to perform final classification calculations on the data to obtain the probability of predicting the label for each sample.

We set Epoch to be 2000, Patience step regarding early stopping to be 12.

6. Evaluation metrics

6.1. Confusion betrics

Our research uses Confusion Metrics to evaluate our convolutional network (Table 1). The confusion metrics show the correct and incorrect prediction of a computer model. From the confusion matrix, we obtain precision, recall, and F1-score for each class, the overall accuracy, and macro and weighted averages of the metrics.

Table 1. Confusion matrix.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Precision is the ratio of true positives (TP) to the sum of true positives and false positives (FP) for a particular class. It measures how many of the samples predicted as positive are actually positive. A high precision indicates that the model correctly identifies positive samples, while a low precision means that the model predicts too many false positives.

Recall is the ratio of true positives to the sum of true positives and false negatives (FN) for a particular class. It measures how many of the actual positive samples are correctly predicted as positive. A high recall indicates that the model is correctly identifying positive samples, while a low recall means that the model is missing many positive samples.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The F1-score is the harmonic mean of precision and recall. It combines precision and recalls into a single metric that balances their trade-offs. The F1 score ranges from 0 to 1, with 1 being the best possible score. A high F1 score indicates that the model performs well in precision and recall.

Accuracy: Accuracy is the ratio of the number of correct predictions to the total number of predictions made by the model. It measures the overall performance of the model. A high accuracy indicates that the model is correctly predicting most of the samples.

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

$$= \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Macro average: Macro average is the average of the evaluation metric calculated for each class. It treats each class equally, regardless of the number of samples in each class.

Weighted average: The weighted average of the evaluation metric is calculated for each class. It considers the **number of samples in** each class, giving more weight to the evaluation metric of classes with more samples.

6.2. Evaluation metrics results description

The following results are obtained using a train-test ratio of 80:20. Based on the provided confusion matrix, we can analyze the performance of a model trained to classify different types of coins. Here are some observations we can make (Figure 3, 4).

The model has an overall accuracy of 0.82, meaning it correctly classified 82% of the samples.

The macro-average F1-score is 0.79, which indicates that the model is performing reasonably well across all classes, but there is room for improvement. The weighted-average F1-score is 0.84, which considers the number of samples in each class and suggests that the model performs better in classes with more samples.

The precision for the "fifty coins" and "hundred coins" classes is high (0.99 and 0.94, respectively), indicating that the model is correctly identifying most of the samples for these classes. However, the precision for the "five coins" and "ten coins" classes is low (0.35 and 0.87, respectively), suggesting that the model is predicting too many false positives for these classes.

The recall for the "five coins" class is high (0.97), indicating that the model is correctly identifying most of the actual "five coins" samples. However, the recall for the "ten coins" class is low (0.56), suggesting that the model is missing many actual "ten coins" samples.

DOI: 10.54254/2755-2721/14/20230795

The F1-score for the "five coins" class is the lowest among all classes (0.51), indicating that the model performs poorly regarding precision and recall. The F1-score for the "hundred coins" class is the highest among all classes (0.97), indicating that the model performs well in terms of precision and recall.

Overall, the model seems to be performing better on classes with more samples ("fifty coins", "hundred coins", and "twenty-five coins") compared to classes with fewer samples ("five coins" and "ten coins"). The model may benefit from more training data and possibly some class balancing techniques to improve its performance on the minority classes.

	precision	recall	f1-score	support
0	0.99	0.87	0.92	120
1	0.35	0.97	0.51	40
2	0.94	1.00	0.97	125
3	0.87	0.56	0.68	121
4	0.98	0.79	0.88	120
accuracy			0.82	526
macro avg	0.83	0.84	0.79	526
weighted avg	0.90	0.82	0.84	526

Figure 3. Confusion matrix data.

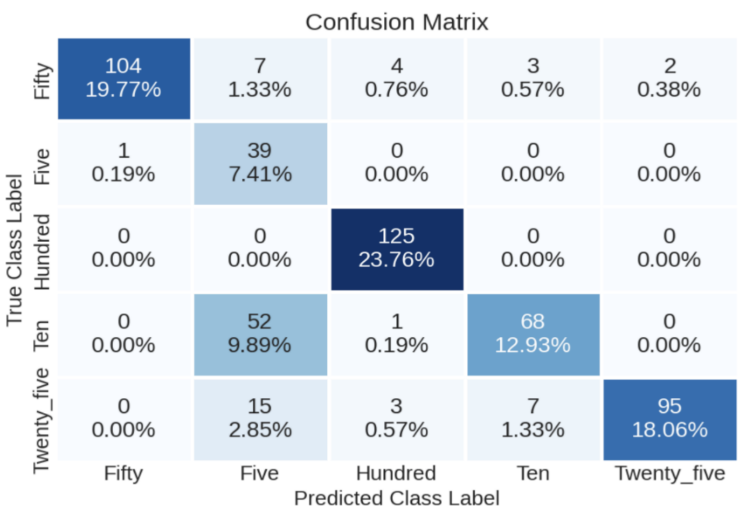

Figure 4. Confusion matrix plot.

7. Experimental results

Our model could recognize coins accurately after training, as shown in Figure 5.

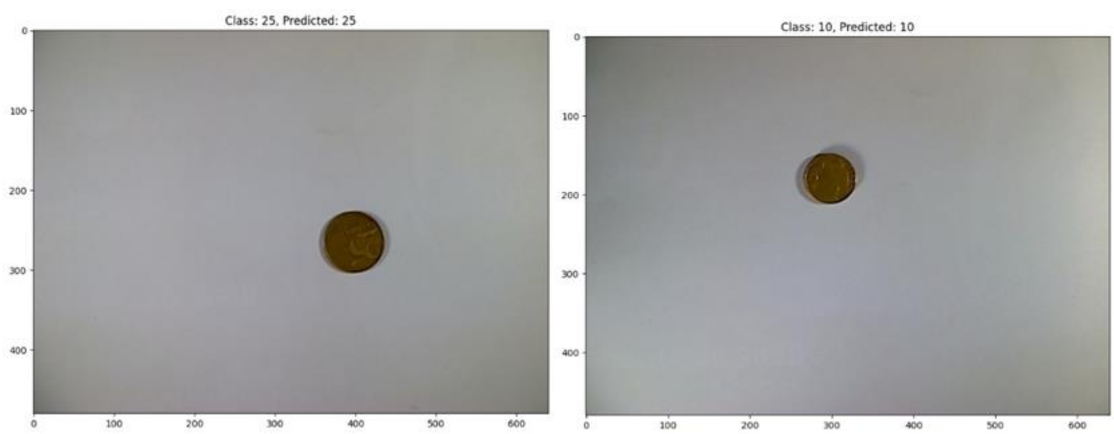

Figure 5. Samples of prediction result.

DOI: 10.54254/2755-2721/14/20230795

To test the stability of our model, we decided to conduct more trials, each with a different train-test ratio, as listed in Table 2. We divided 3059 images into two parts: the training and validation sets. The accuracy of the results first increased and then decreased as the proportion of the training set increased from 75% to 90%. When the train-test ratio is 75:25, the best accuracy reached about 96.73%. When the train-test ratio is 80:20, the best accuracy reached about 98.04%, which is the same for a train-test ratio of 85:15. When the train-test ratio is 90:10, the highest accuracy reached about 97.39%. We also noticed that the program generally enters a stable state after 35-50 epochs and ends early.

Table 2. Table of validation accuracy for different train-test ratios.

Training Set	75%	80%	85%	90%
Test Set	25%	20%	15%	10%
Epoch	52	49	36	37
Val_Accuracy	0.9673	0.9804	0.9804	0.9739

The accuracy results in each step are shown in the following four images (Figure 6, 7, 8, 9).

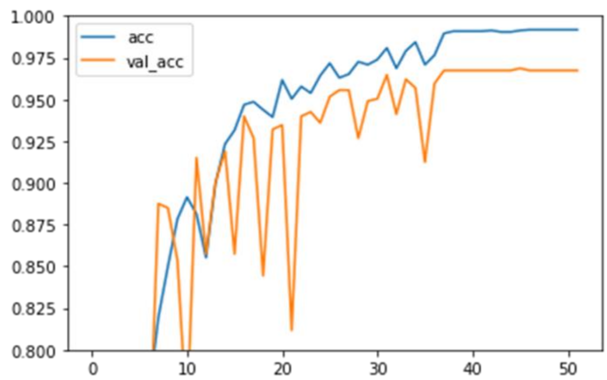

Figure 6. Accuracy curve with 75% as training set & 25% as test set.

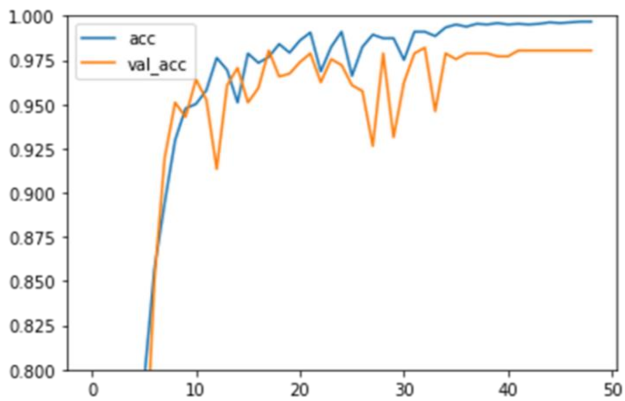

Figure 7. Accuracy curve with 80% as training set & 20% as test set.

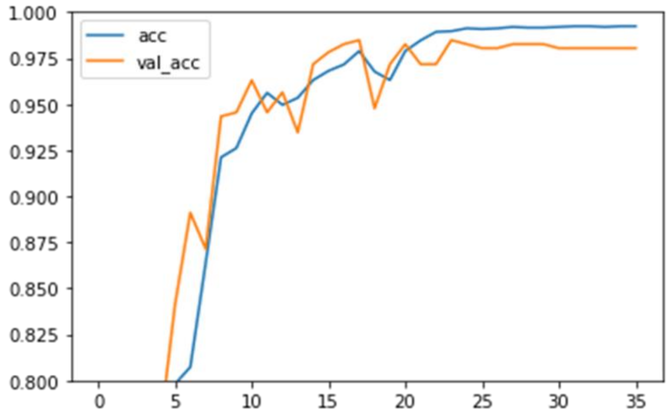

Figure 8. Accuracy curve with 85% as training set & 15% as test set.

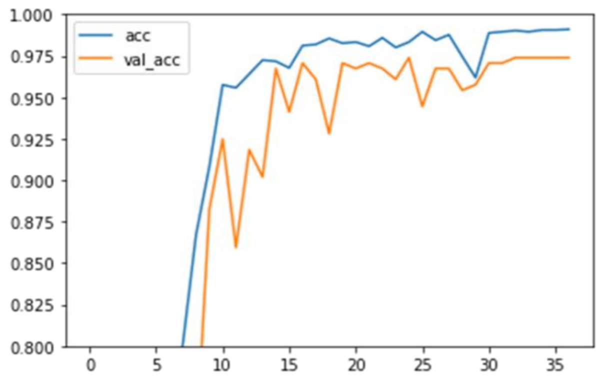

Figure 9. Accuracy curve with 90% as training set & 10% as test set.

8. Discussions & limitations

After observing that our model reached an overall highest accuracy of 98 percent, we can conclude that CNN (convolutional neural networks) is a highly effective algorithm for image processing and classification.

As the number of parameters in a neural network increase with the addition of more layers, the process of training a model can become heavy computationally and sometimes not feasible. Images have high dimensionality and usually contain many parameters since each pixel is considered a feature. This makes tuning all these parameters difficult and time-consuming. However, CNN can significantly facilitate this task because the multiple convolution filters found in each convolutional layer can scan the entire feature matrix of the images and carry out dimensionality reduction. In our model, each convolutional layer is also followed by a Max Pooling layer where the images' spatial dimensionality is further reduced without losing the essential features. This condenses and reduces the computation needed for the following layers of the neural network. Therefore, CNN can reduce the number of parameters without losing the images' quality or information.

Despite the high accuracy, there exist limitations to our CNN model. One notable limitation is the time required to train it. Our dataset consists of over 3000 different images. Therefore, during our experiments, training only 20 epochs took at least one and a half hours. Because of this highly time-consuming training process, we did not train our model over 50 epochs. However, we infer that our CNN model can likely achieve higher accuracy with more epochs of training.

9. Conclusion

We construct the model using the Keras library to create a convolutional neural network (CNN) architecture, a common approach for image classification tasks.

The model is trained using a training set and validated using a separate validation set. The model's performance is evaluated using several metrics: accuracy, precision, recall, and F1 score. The model achieves the highest accuracy of around 98% on the validation set. This indicates the model can accurately classify most images in the validation set. The precision, recall, and F1 score metrics are also high, indicating that the model can balance correctly identifying positive and negative samples.

Overall, the model's performance appears to be very good, with high accuracy and other metrics that suggest the model can accurately classify images of Brazilian coins. We, therefore, conclude that CNN is an apt and suitable model for image classification. However, it is essential to note that the model's performance may vary depending on the specific application and the characteristics of the images being classified.

Future Work: One major challenge that we are currently faced with is that our CNN model's training process is extremely time-consuming. Future experiments and modifications are expected to find a solution to reduce the time required for the model's training process. Once the time needed to train the model decreases, the model should be trained for more epochs since for larger numbers of epochs, the accuracy tends to increase. The learning rate, which is set to 0.001 in our experiment, may also be modified and tuned to optimize the model's performance. Additionally, the current dataset size may be increased to give CNN more available images, improving performance. This is because CNN needs a lot of training data to automatically learn features from the images.

References

[1] Dabic, S. (2011, September 7). WO2012036956A1 - Coin Identification Method and apparatus. Google Patents. Retrieved March 3, 2023, from https://patents.google.com/patent/WO2012036956A1/en
[2] Pham, T.D. et al. (2022) Deep learning-based detection of fake multinational banknotes in a cross-dataset environment utilizing smartphone cameras for assisting visually impaired individuals, MDPI. Multidisciplinary Digital Publishing Institute. Available at: https://www.mdpi.com/2227-7390/10/9/1616 (Accessed: March 10, 2023).
[3] C. Park, S. W. Cho, N. R. Baek, J. Choi and K. R. Park, "Deep Feature-Based Three-Stage Detection of Banknotes and Coins for Assisting Visually Impaired People," in IEEE Access, vol. 8, pp. 184598-184613, 2020, doi: 10.1109/ACCESS.2020.3029526.
[4] Yufeng, X., & Yan, W. Q. (2021). Fast-moving coin recognition using deep learning. Multimedia Tools and Applications, 80(16), 24111-24120. https://doi.org/10.1007/s11042-021-10857-5
[5] https://www.kaggle.com/datasets/lgmoneda/br-coins