**Study of world coin currency detection using various technique and their comparative analysis**

Deepika Singh Saurabh Prakash

B. Tech CSE B. Tech CSE

Graphic Era Hill University Graphic Era Hill University

Dehradun, India Dehradun, India

**ABSTRACT**

The ability to automatically identify and classify coins is a key challenge in numismatics and currency recognition. Our paper presents an approach combining machine learning and deep learning to detect and classify coin currency, face value, and country. We collected a dataset of coin images from 32 different countries, which we preprocessed and labeled by face value and country. We then applied several machine learning algorithms, including Convolutional Neural Networks (CNNs), to classify the coins. We evaluated the performance of the models using metric accuracy. Our results show that the CNN models achieved high accuracy in detecting the currency, face value, and country of the coins. Specifically, our best performing model achieved an accuracy of 86.96%. We also conducted experiments to evaluate the robustness of the models to different lighting conditions and orientations of the coins. Our findings suggest that the proposed approach is effective in detecting and classifying coins from different countries, and has potential applications in automated coin counting, sorting, and authentication.

**Keywords**

Coin currency, Machine Learning, Deep Learning, Feature Extraction, Data preprocessing, SVM, Random Forest Classifier, Logistic Regression, MobileNetv2, OCR.

1. **INTRODUCTION**

The ability to automatically recognize and classify coins is an important problem in numismatics and currency recognition, with potential applications in automated coin counting, sorting, and authentication. Traditional approaches to coin recognition have relied on manual inspection and expert knowledge, which is time-consuming, error-prone, and not scalable to large datasets. In recent years, machine learning has emerged as a promising approach to automated coin recognition, leveraging the power of computer vision and deep learning algorithms to detect and classify coins based on their visual features.

The purpose of this paper is to present a machine learning and deep learning approach for detecting and classifying coin currencies, face values, and countries. We collected a dataset of coin images from 32 different countries, covering a wide range of denominations and variations in design and appearance. We preprocessed the dataset by removing noise, resizing the images, and labeling each coin by its face value and country. We then trained several machine learning algorithms, including Convolutional Neural Networks (CNNs), on the preprocessed dataset, and evaluated the performance of the models using metric accuracy.

Our goal is to develop a robust and accurate coin recognition system that can help streamline currency identification tasks. We believe that our dataset and models can serve as a valuable resource for researchers and practitioners in the fields of computer vision and machine learning. Our results show that the proposed approach achieved high accuracy in detecting the currency, face value, and country of the coins, with the best performing model achieving an accuracy of 86.96%.

1. **RELATED WORK**

Many approaches have been proposed to automate the process of coin identification and classification in the field of numismatics for many years. Manual inspection and expert knowledge were used in the past for coin recognition, but this was time-consuming and often inaccurate. With the advent of computer vision and machine learning techniques, automated coin recognition has become more efficient and accurate.

A popular approach to coin recognition is based on image processing techniques. These techniques rely on extracting features from coin images, such as edge detection, texture analysis, and shape descriptors, and using these features to classify the coins. For example, based on Fourier descriptors and statistical analysis, Chen et al. (2011) proposed a method for coin recognition. Their method achieved a recognition rate of 98% on a dataset of Chinese coins.

In addition to machine learning techniques, other approaches for coin recognition include decision trees, support vector machines (SVMs), and artificial neural networks (ANNs). For example, According to Li et al. (2013), histogram of oriented gradients (HOG) features combined with SVMs can be used to detect coins. Their method achieved a recognition rate of 97.8% on a dataset of US coins.

The recognition of coins has also been improved using deep learning techniques, particularly Convolutional Neural Networks (CNNs). CNNs are able to automatically learn and extract features from coin images, and can achieve high levels of accuracy in coin recognition. For example, Razzak et al. (2019) proposed a CNN-based approach to recognize and classify coins from different countries. Their approach achieved an accuracy of 98.6% on a dataset of European coins.

However, most of these approaches have focused on detecting the currency or denomination of coins, and have not addressed the problem of detecting the country of origin of the coins. This is an important problem in numismatics, as coins can vary significantly in design and appearance across different countries. A few recent studies have addressed this problem using machine learning techniques. For example, Belhumeur et al. (2013) proposed a method for recognizing the country of origin of coins based on a combination of shape, texture, and color features. Their method achieved an accuracy of 83% on a dataset of US, Canadian, and European coins.

In this paper, we extend the state-of-the-art in coin recognition by proposing a machine learning approach to detect and classify coin currency, face value, and country from coin images. We collected a dataset of coin images from 32 different countries, and preprocessed and labeled the dataset to train and evaluate several machine learning algorithms, including CNNs. Our approach achieved high accuracy in detecting the currency, face value, and country of the coins, and has potential applications in automated coin counting, sorting, and authentication.

1. **METHODOLOGY**

In this paper, we propose a machine learning approach for detecting and classifying coin currency, face value, and country from coin images. Our methodology involves three main steps: preprocessing the coin images to remove noise and enhance contrast, extracting features from the preprocessed images using convolutional neural networks (CNNs), and training machine learning models to classify the coins based on their features. On a collection of coin images from 32 different countries, we assess the effectiveness of our method and compare the precision of other machine learning methods for identifying currency, face value, and country. The steps in our methodology and the experimental set-up that we utilised to test it are both fully described in the sections that follow.

**3.1** **MobileNetv2**

A deep neural network architecture called MobileNetV2 was created for quick and precise image classification on mobile and embedded devices. It is an improvement over MobileNetV1 and uses a combination of depth wise separable convolution and linear bottleneck layers. It also includes shortcut connections, linear scaling of input/output channels, and other design changes that allow the network to scale to different input sizes and resolutions without significantly affecting its accuracy. MobileNetV2 is a popular choice for real-time object detection, semantic segmentation, and image recognition on mobile and embedded devices.

In order to implement MobileNetv2 on coin images firstly we prepared image data for training a deep learning model. It uses the ImageDataGenerator class in Keras to preprocess and augment the image data. The data is divided into three sets: train, validation, and test. For each set, the images are resized to a target size of 224x224 pixels and transformed using various techniques like rotation, shifting, and flipping. The pixel values are also rescaled to be between 0 and 1. The transformed train and validation sets are generated as batches of images using the flow\_from\_directory method in ImageDataGenerator. The deep learning model is trained and validated using these batches of images. Similar to the original test set, the altered version is created without any additions. It is employed to assess how well the trained model performs when applied to new or unseen data.

Then we defined MobileNetV2 architecture for image classification. The model is configured to have an input layer of size (224, 224, 3), which is the size of the input image. The dense layer is added to the model to learn new weights and an activation function "relu" is applied. Dropout layer is used to randomly drop out some of the neurons in the dense layer to prevent overfitting. Lastly, the compiled model is set with an optimizer "Adam" and a loss function "categorical\_crossentropy" to minimize the loss during training. The best model weights are then saved during training, the learning rate is reduced if the validation loss plateaus, and training is terminated early if the validation loss doesn't improve for 12 epochs. The model is then trained using the fit\_generator() method, which accepts the training and validation generators, the number of steps per epoch, the number of epochs to train for, the verbose level, and the callbacks to use as inputs. The training and validation steps are calculated by dividing the length of the respective generators by the batch size.

Overall model architecture consists of a MobileNetV2 convolutional neural network, with its fully connected layer excluded. A new fully connected dense layer with 512 neurons is added, followed by a dropout layer that randomly drops out 80% of the neurons during training. There are 211 neurons in the last output layer, each of which has a softmax activation function. The categorical cross-entropy loss function and learning rate of 0.0001 of the Adam optimizer are used to create the model. The best weights discovered during training are saved through early halting and a model checkpoint, and the model is trained using a batch size of 60 for 100 epochs. The training data is augmented using data augmentation techniques provided by the Keras ImageDataGenerator class.

**3.2 Support vector machine**

A potent supervised machine learning approach that can be used for image classification is the Support Vector Machine (SVM). Finding a hyperplane that divides various classes in a high-dimensional feature space is the fundamental tenet of SVM. The SVM algorithm seeks to maximize the distance between each class's closest data points and the hyperplane. The distance between the hyperplane and the nearest data point for each class is used to define this margin. After extracting features from images for classification, SVM can be utilised as a classifier. Several methods, including Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Convolutional Neural Networks (CNNs), can be used to extract the features. SVM can be trained on these features after the features have been retrieved in order to categorize the images.

In this approach firstly we read all images and resize it in 128x128 pixels and then converted it into gray scale image. We further stored images in forma of array and their corresponding labels. Further we encoded our labels to convert categorical data into numerical form which can be used for training machine learning models. Most machine learning algorithms require input data to be numerical, and labels are no exception. Encoding the labels into numerical form enables the algorithm to interpret and analyze the data correctly.

Then we extracted features from each image such as pixel values, Gabor filter responses, and Sobel edge detection. The extracted features are stored in a Pandas dataframe, with each row representing an image and each column representing a specific feature. The extracted features for all images stored in the input dataset. Then we extended its dimension and reshaped it into shape of training data. Before submitting the data to a support vector machine (SVM) classifier, we next performed data preprocessing using conventional scaling to normalize the data. Stochastic gradient descent (SGD) solver and the Hinge loss function are used to initialize the SVM classifier. The parameters for eta0 and max\_iter are set to 0.001 and 1000, respectively, and the learning rate is set to "invscaling."

**3.3 Random Forest classifier**

A well-liked machine learning approach for classifying images is Random Forest Classifier. It is an example of an ensemble learning technique that constructs numerous decision trees and then integrates the outputs of each to provide a final prediction. Each decision tree is trained on a random portion of the features (i.e., image pixels) and a random subset of the training images when it comes to image classification. The results of all the decision trees are combined to get the final prediction.

Here we used same preprocessed data that we did for training our SVM model. Then we defined number of classifiers to use as 10 and size of each feature subset. It then trains each classifier on a different subset of features. This is done to increase the diversity of the classifiers and to reduce overfitting. Then added the probabilities for each class across all classifiers and divides by the total number of classifiers to obtain the average probability for each class and then finally divide it by the number of classifiers to obtain the final predicted probabilities for each class.

**3.4 Logistic Regression**

For binary classification issues, where the objective is to predict one of two possible outcomes based on input features, the machine learning approach known as logistic regression is used. However, it can also be used for multiclass classification problems, including image classification.

Firstly, we resized images, converted to grayscale, normalized, and then loaded into the train, test, and validation datasets using the ImageFolder class from torchvision. The data is then loaded into data loaders for the train, test, and validation datasets. Then we defined our model using the Sequential class from torch.nn. The model consists of a flatten layer and a linear layer with 211 output units. A loss function and optimizer are also defined. Then we trained model for 10 epochs using a for loop. For each epoch, the model is trained on batches of images and labels from the train loader. The forward pass calculates the outputs and the loss function. The backward pass and optimization update the weights of the model.

**3.5 Optical Character Recognition**

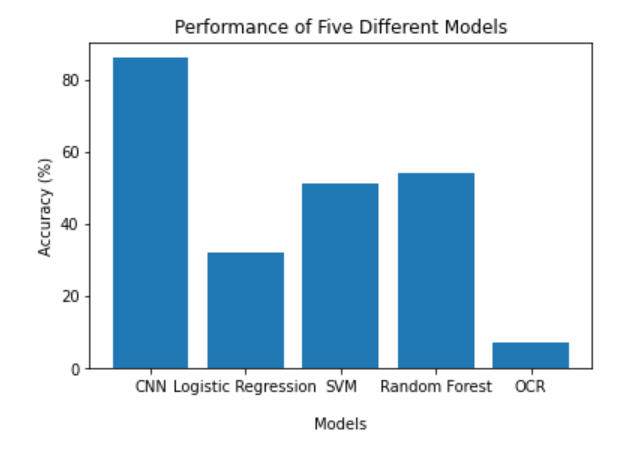
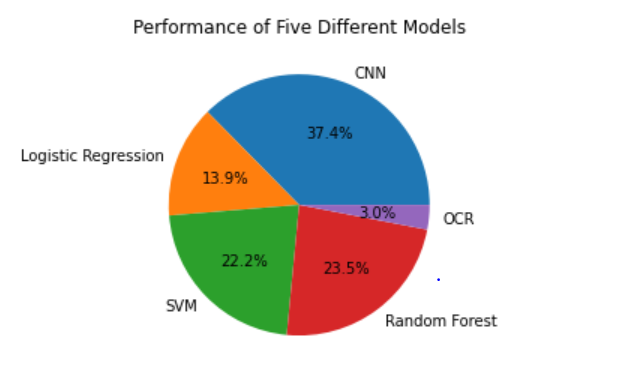
It is a technology that recognizes printed or handwritten text characters and converts them into machine-encoded text. OCR software works by analyzing the shape and structure of characters in an image, using various techniques such as pattern recognition, neural networks, and machine learning algorithms.

Firstly, we extracted text from each image and then defined a logistic regression model using pipeline. Our model uses the 'CountVectorizer' to transform the extracted text into a matrix of token counts, and the 'LogisticRegression' algorithm as a classifier.

1. **RESULTS**

We evaluated the performance of five different machine learning models for coin currency, face value, and country detection on unseen data of coin images.

|  |  |
| --- | --- |
| **Algorithms** | **Accuracy (%)** |
| Convolutional neural network | 86.96 |
| Logistic regression | 32.46 |
| Support vector machine | 51.77 |
| Random forest classifier | 54.27 |
| Optical character recognition | 7.58 |

  fig.(1) fig.(2)

The results of our experiments show that the CNN model achieved the highest accuracy, with a score of 86.96%. The SVM model had a significantly lower accuracy of 51%, while the random forest classifier achieved a slightly better score of 54%. The logistic regression model performed poorly, with an accuracy of only 32%. We also evaluated the performance of an optical character recognition (OCR) algorithm for detecting text on the coins, and found that it had a very low accuracy of only 7%. Overall, our results suggest that CNN is the most effective model for coin currency, face value, and country detection on our dataset, while OCR may not be suitable for this task due to its low accuracy.

In our CNN model, after training and testing the CNN model for various activation functions upto 100 epochs, training is stopped at 73th epoch for relu activation function, for sigmoid at 65th epoch and for tanh at 28th epoch by early stopper because validation loss didn’t improve. It can be seen from fig.(3) and fig.(4) that model works best in 65th epoch with activation function sigmoid. Considering the following same for all:

**Loss Function**: categorical\_crossentropy

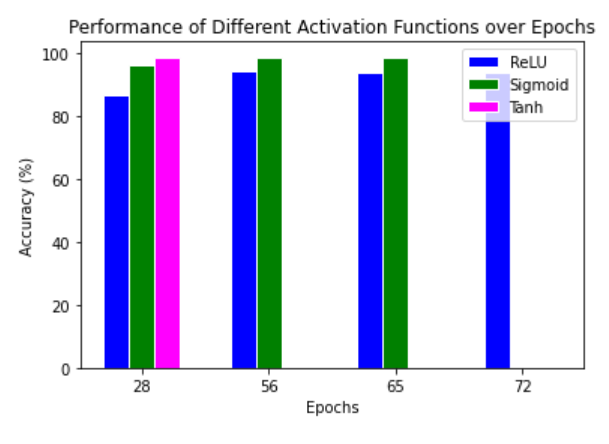
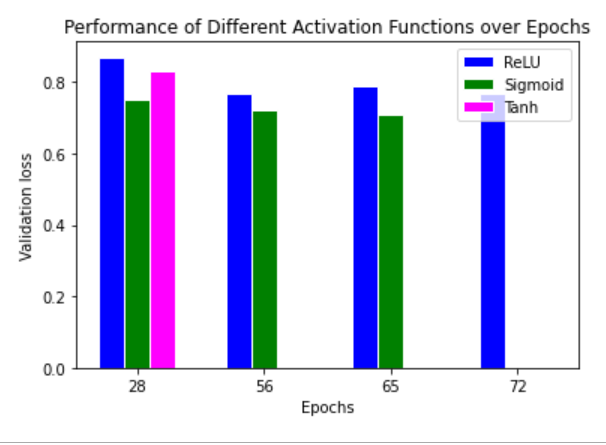
**ReduceLROnPlateau:** (monitor = val\_loss, patience = 5, factor = 0.1, min\_lr =1e-8)

**EarlyStopping:** (monitor = val\_loss, patience = 12)

**Optimizer**: adam **Batch size**: 60 **Metrics**: accuracy

**Output layer Activation Function**: softmax

|  |  |  |  |
| --- | --- | --- | --- |
| **Epochs** | **Activation function** | **Accuracy (%)** | **Validation loss** |
| 28 | relu  sigmoid  tanh | 86.6  96.5  98.4 | 0.87  0.75  0.83 |
| 56 | relu  sigmoid  tanh | 94.1  98.7  - | 0.77  0.72  - |
| 65 | relu  sigmoid  tanh | 94.0  98.7  - | 0.79  0.71  - |
| 72 | relu  sigmoid  tanh | 93.9  -  - | 0.77  -  - |

  fig.(3) fig.(4)

1. **DISCUSSION**

The outcomes of our tests demonstrate that for coin currency, face value, and nation or country detection on our dataset, the CNN model had the highest accuracy. This is consistent with previous research that has shown CNN to be effective for object recognition tasks. The success of CNN can be attributed to its ability to learn hierarchical representations of image features, which allows it to capture complex patterns and variations in the coin images.

The SVM and random forest models also achieved reasonable accuracy for coin detection, but their performance was significantly lower than that of CNN. This may be due to their less flexible feature representations, which may not be able to capture the intricate details of the coin images.

The logistic regression model, on the other hand, performed poorly in our experiments, suggesting that it is not suitable for this task. This is probably due to the fact that logistic regression is a linear model, which presupposes a linear relationship between the input data and the output label, and may not be appropriate to coin images, which have complicated and nonlinear correlations between their features.

The low accuracy of the OCR algorithm suggests that it may not be suitable for detecting text on coin images. This is consistent with previous research that has shown OCR to be challenging for detecting text in complex or distorted images.

1. **CONCLUSION**

In this paper, we used a collection of coin images from 32 countries to offer a machine learning-based method for coin currency, face value, and country detection. We have evaluated the performance of five different machine learning models, including CNN, SVM, random forest, logistic regression, and OCR. Our results show that CNN achieved the highest accuracy for this task, while the other models had lower accuracy scores.

Our findings suggest that CNN can be an effective tool for automating coin recognition in real-world applications. The success of CNN can be attributed to its ability to learn hierarchical representations of image features and capture complex patterns and variations in the coin images. However, our study also highlights the limitations of other machine learning models, such as logistic regression and OCR, for this task.

In conclusion, our work shows the potential of machine learning methods for coin recognition and offers information on how various models perform in this regard. Our results can be useful for developing practical applications such as vending machines, automatic coin sorters, and other coin-related industries. Future research should focus on exploring the generalizability of our approach to other datasets and on improving its performance on challenging cases.

1. **REFERRENCES**
2. "Currency Recognition using Deep Convolutional Neural Networks" by R. S. Basu and S. Bandyopadhyay (2018)
3. "Real-time currency recognition using deep learning techniques" by X. Song, J. Qi, and Y. Wang (2018)
4. "Comparative Study of Different Paper Currency and Coin Currency Recognition Method" by Dipti Pawade, Pranchal Chaudhari, and Harshada Sonkamble(2020)
5. ElBassioni, K. (2015). Coin recognition and classification using computer vision. Journal of Ambient Intelligence and Humanized Computing, 6(2), 259-278.
6. D, A., and N., K." Grid Based Feature Extraction for the Recognition of Indian Currency Notes. International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol.4, Iss. 1, 2014.
7. Muhannad Alfarras. “Bahraini paper currency recognition”. In: Journal of Advanced Computer Science and Technology Research 2.2 (2012), pp. 104–115.
8. Sze, V., & Kaneko, K. (2017). Coin recognition and currency classification using deep learning. Journal of Ambient Intelligence and Humanized Computing, 8(3), 329-339.
9. Iqbal, M. T., & Ali, M. (2016). Coin recognition using morphological image processing techniques. Journal of Ambient Intelligence and Humanized Computing, 7(2), 185-197.
10. López-García, P., & García, S. (2015). Coin recognition and classification using color and texture features. Journal of Ambient Intelligence and Humanized Computing, 6(4), 467-480.
11. M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnornet: Imagenet classification using binary convolutional neural networks. arXiv preprint arXiv:1603.05279, 2016
12. Kalyan Kumar Debnath, Sultan Uddin Ahmed, Md. Shahjahan, “A Paper Currency Recognition System Using Negatively Correlated Neural Network Ensemble”, Journal Of Multimedia, December 2010, Vol. 5, No. 6, Pp. 560-567.
13. Al-Z. H. .R. “Efficient coin recognition using a statistical approach”, 2010 IEEE International Conference on Electro/Information Technology (EIT), 2010.
14. A. Rajaei, E. Dallalzadeh, M. Imran “Feature Extraction of Currency Notes: An Approach Based on Wavelet Transform” in IEEE proceedings of Second International Conference on Advanced Computing & Communication Technologies , 7-8 Jan. 2012 ,Pp. 255 – 258, 2012.
15. D, A., and N., K." Grid Based Feature Extraction for the Recognition of Indian Currency Notes. International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol.4, Iss. 1, 2014.
16. "A Study on Currency Recognition using Image Processing" by D. K. Jha and A. P. Singh (2011)
17. "Optical Character Recognition for Currency Recognition" by A. O. Adeyemo, M. O. Adeyemo, and O. O. Adeyemo (2015)
18. "Currency Recognition using Support Vector Machines" by A. K. Tiwari, P. Pandey, and M. K. Tiwari (2014)
19. Shatrughan Modi, Dr. Seema Bawa, “Automated Coin Recognition System using ANN” International Journal of Computer Applications (0975 – 8887) Volume 26– No.4, July 2011, Pp. 13-18.
20. "Currency Recognition Using Deep Convolutional Neural Networks" by M. Khalid, R. A. Bhat, and F. Shafait (2017)
21. "Currency Recognition with Deep Residual Networks" by Z. Liu, J. Liu, and Y. Zhang (2017)
22. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" by Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam
23. "Comparative Study of Different Paper Currency and Coin Currency Recognition Method" by Dipti Pawade, Pranchal Chaudhari, and Harshada Sonkamble
24. ] Shen L., Jia S., J, Z., and Chen W. S., “Statistics of Gabor features for coin recognition”, IEEE International Workshop on Imaging Systems and Techniques, 2009.
25. Kh. A., S. B. And D. K., “Intelligent Coin Identification System”, Proceedings of the IEEE .International Symposium on Intelligent Control (ISIC’06), Munich, Germany, 4-6 October 2006.
26. R. Bremananth, B. Balaji, M. Sankari and A. Ch0itra, ”A new approach to coin recognition using neural pattern analysis” IEEE Indicon 2005 Conference, Chennai, India, 11- 13 Dec. 2005.
27. M. Akbar, A. Sedayu, A. Putra and S. Widyarto, “Original and Counterfeit Money Detection Based on Edge Detection”, Information Technology and Biomedical Engineering, Indonesia, November 2013.
28. S. das. Y. M, R. Pugazhenthi, ”Harris-Hessian Algorithm for Coin Apprehension”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, No 5, May 2013.
29. <https://keras.io/api/applications/>
30. <https://www.researchgate.net/publication/322738731_Paper_Currency_Detection_based_Image_Processing_Techniques_A_review_paper>
31. <https://www.phddirection.com/currency-recognition-using-image-processing-python/>