# Modern Currency Recognition: A Deep Learning Approach

Ajay Sasane

Department of Science and Engineering, Manchester Metropolitan University, Manchester, UK

23681593@stu.mmu.sc.uk

***Abstract*— In this study, a deep learning approach was employed to classify Indian currency notes using a convolutional neural network (CNN). The dataset, obtained from Kaggle, consisted of images of various denominations of Indian currency notes. The data was divided into training, validation, and test sets, and appropriate preprocessing techniques, including resizing and tensor transformation, were applied. A custom CNN architecture was designed, incorporating multiple convolutional and max-pooling layers, followed by fully connected layers. The model was trained using the Adam optimizer and cross-entropy loss function over 100 epochs. During training, data augmentation techniques were utilized to enhance the robustness of the model. The performance of the model was evaluated using accuracy metrics on the validation and test datasets. The results demonstrated that the model achieved a high level of accuracy in classifying the currency notes. Training and validation loss curves, as well as accuracy curves, were plotted to monitor the learning process and identify potential overfitting or underfitting issues. Additionally, a confusion matrix was generated to visualize the model's performance across different classes. The model was further tested using an interactive interface that allowed users to upload images and receive predictions with confidence scores. The study highlights the effectiveness of deep learning techniques in image classification tasks, particularly in the context of currency note recognition. Future work may involve the exploration of more complex architectures, larger datasets, and additional data augmentation methods to further improve model performance and generalization capabilities.***

***Keywords*—Deep Learning, Convolutional Neural Network (CNN), Image classification, Currency Note Recognition, Model Evaluation, Adam Optimizer, Counterfeit detection**

## I. Introduction

### *A. Background*

The accurate classification and recognition of currency notes are vital tasks in the financial and banking sectors globally. These tasks play a crucial role in mitigating issues related to counterfeit currency, which can have severe economic consequences. In India, the government initiated a significant monetary reform on November 8, 2016, known as demonetization. This reform aimed at curbing the proliferation of fake currency notes, reducing corruption, and promoting digital transactions. The initiative involved the withdrawal of 500 and 1,000 rupees banknotes from circulation, necessitating the introduction of new currency notes with enhanced security features (Reserve Bank of India, 2017).

In this study, a convolutional neural network (CNN) has been developed to classify Indian currency notes, leveraging deep learning techniques. The dataset, comprising images of various denominations, including those introduced post-demonetization, has been utilized. Advanced preprocessing, model training, and evaluation methods were employed to achieve high accuracy and robustness in currency classification.

### *B. Research gap*

Despite the critical importance of currency recognition, many existing systems continue to rely on traditional image processing techniques that often lack robustness against variations in note conditions, lighting, and sophisticated counterfeit strategies (Jetir, 2023). The advancements in machine learning, particularly in the field of deep learning, present an opportunity to develop more sophisticated and accurate models for currency recognition (LeCun, Bengio, and Hinton, 2015). However, there exists a significant gap in the literature regarding the application of deep learning techniques for the classification of Indian currency notes, especially in the context of the post-demonetization era where new notes with updated security features are in circulation.

The implications of demonetization further underscore this research gap. The withdrawal of old notes and the subsequent introduction of new ones necessitated the development of updated methods to accurately identify the newly issued currency (Brookings, 2017). This transition period saw an increased demand for automated systems capable of quickly and accurately validating currency notes in banks, ATMs, and other financial institutions, thereby reducing the risk of counterfeit transactions. The development of reliable and efficient currency recognition systems, as pursued in this study, is essential not only for countering counterfeit currency but also for streamlining the cash handling process in the wake of significant monetary policy changes.

This study addresses these gaps by leveraging deep learning techniques to develop a convolutional neural network (CNN) for the classification of Indian currency notes. The research utilizes a dataset comprising images of various denominations, including those introduced post-demonetization, and employs advanced preprocessing, model training, and evaluation methods to achieve high accuracy and robustness in currency classification.

## II. Related work

### *A. Traditional Methods*

The recognition and classification of currency notes have been widely explored in the realms of image processing and computer vision. Traditional methods for currency recognition typically utilize handcrafted features and rule-based algorithms. Techniques such as edge detection, histogram analysis, and template matching have been employed to identify distinctive features of currency notes. However, these methods often exhibit limited robustness against variations in note conditions, lighting, and counterfeit strategies.

### B. *Advancements with Deep Learning*

The advent of machine learning, and more specifically deep learning, has introduced more sophisticated models capable of achieving superior performance in image classification tasks. Convolutional neural networks (CNNs) have shown exceptional success in applications such as object detection, facial recognition, and medical image analysis (LeCun, Bengio, and Hinton, 2015). CNNs excel by automatically learning hierarchical feature representations from raw pixel data, enhancing their ability to generalize across various conditions and variations.

### C. *Deep Learning in Currency Recognition*

Recent advancements in currency recognition have demonstrated the significant potential of deep learning techniques to enhance accuracy and robustness. Bhurke et al. utilized hue, saturation, and value (HSV) color models from images of multiple currencies, employing template-matching for recognition. Their method showed reliable performance across various currencies, including USD, EUR, and INR, highlighting the versatility of deep learning in currency classification (Bhurke et al.)

Similarly, Pham et al. integrated convolutional neural networks (CNNs) to classify banknotes from six different national currencies, achieving high accuracy by considering the size characteristics of the banknotes. This approach underscores the effectiveness of CNNs in currency recognition, suggesting that deep learning methodologies can significantly improve the performance of automated currency recognition systems (Pham et al.)

## III. DATASET AND PRE-PROCESSING

### A. *Dataset*

The dataset used in this study was sourced from Kaggle (Sahani, 2020) and comprises high-resolution images of Indian currency notes. The dataset contains images of various denominations, including ₹10, ₹20, ₹50, ₹100, ₹200, ₹500, and ₹2000 notes. The dataset is divided into two main folders: Train and Test. Each folder contains subfolders representing the different classes of currency notes (Fig. 1.). The Train folder consists of images used for training the model, while the Test folder comprises images used for evaluating the model's performance.


Fig. 1. Currency classes

### B. *Preprocessing*

Preprocessing steps were implemented to prepare the images for input into the convolutional neural network (CNN). The preprocessing pipeline included the following steps:

**Resizing:** Each image was resized to a fixed dimension of 224x224 pixels. This step ensured uniformity in image dimensions, which is crucial for processing by the CNN. The resizing was performed using the 'transforms.Resize' function from the torchvision library.

**Normalization:** The pixel values of the images were normalized to the range [0, 1] by converting the images to PyTorch tensors. This step was achieved using the 'transforms.ToTensor' function, which converts the images from PIL format to tensor format and scales the pixel values accordingly.

**Data Augmentation:** To enhance the robustness of the model, data augmentation techniques were applied to the training images. These techniques included random rotations, horizontal flips, and slight translations. The objective of data augmentation was to simulate various conditions and variations that the model might encounter in real-world scenarios. The 'transforms.Compose' function was used to apply these transformations in sequence.

**Dataset Splitting:** The dataset was split into training, validation, and test sets. The training set comprised 90% of the images from the Train folder, while the remaining 10% formed the validation set. This splitting ensured that the model could be trained and validated effectively, with the test set used exclusively for evaluating the final model performance. The 'random_split' function from the 'torch.utils.data' module was employed for this purpose.

**Data Loading:** Data loaders were created for the training, validation, and test sets to facilitate batch processing during model training and evaluation. The 'DataLoader' class from the 'torch.utils.data' module was utilized, with a batch size of 32 set for all loaders. Shuffling was enabled for the training and validation loaders to ensure that the model does not learn any ordering in the data.

These preprocessing steps ensured that the images were suitably prepared for input into the CNN, enabling efficient training and evaluation of the model.

## IV. PROPOSED METHODOLOGY

### A. *Convolutional Neural Network Architecture*

A custom convolutional neural network (CNN) was designed to classify Indian currency notes. The architecture of the CNN consists of multiple convolutional layers, followed by max-pooling layers and fully connected layers. The network begins with three convolutional layers. Each convolutional layer applies a series of filters to the input image to extract local features such as edges, textures, and patterns. Specifically, the first convolutional layer uses 64 filters of size 3x3 with a stride of 1 and padding of 1, followed by a ReLU activation function. The second convolutional layer utilizes 128 filters of the same size and configuration, also followed by a ReLU activation function. The third convolutional layer comprises 256 filters of size 3x3 with a stride of 1 and padding of 1, with a subsequent ReLU activation function. Max-pooling layers are interspersed between the convolutional layers to reduce the spatial dimensions of the feature maps and retain the most important features. A max-pooling layer with a kernel size of 2x2 and a stride of 2 follows each of the first two convolutional layers, while a max-pooling layer with a kernel size of 4x4 and a stride of 4 follows the third convolutional layer. The output from the final max-pooling layer is flattened and passed through a series of fully connected layers to perform the final classification. The first fully connected layer contains 256 neurons with a ReLU activation function, followed by a second fully connected layer with 256 neurons and a ReLU activation function. The output layer consists of 7 neurons (corresponding to the seven classes of currency notes) with a softmax activation function to output the probabilities for each class.

### B. *Training Procedure*

The training procedure involved several key steps. The preprocessed images were loaded into the model using PyTorch's DataLoader, with the training dataset divided into batches of size 32 to facilitate efficient processing and gradient updates. The cross-entropy loss function was employed to measure the discrepancy between the predicted class probabilities and the true class labels, a suitable choice for multi-class classification tasks. The Adam optimizer, selected for its adaptive learning rate capabilities, was used to train the model with a learning rate set to 0.0003. The training loop iterated over the dataset for 100 epochs. In each epoch, a forward pass was conducted where the input images were passed through the CNN to obtain predicted class probabilities. The cross-entropy loss was then computed using the predicted probabilities and true labels, followed by a backward pass to calculate gradients through backpropagation. The optimizer updated the model parameters, and training loss and accuracy were recorded for each batch. After each epoch, the model was evaluated on the validation dataset, with the same metrics (loss and accuracy) calculated to monitor the model's performance and detect potential overfitting. The model with the highest validation accuracy was saved as the best model to ensure that the best-performing model on unseen data was retained for further evaluation.

### C. *Evaluation and Interactive Prediction Interface*

The final evaluation of the model was conducted on the test dataset, where test accuracy was computed as the average accuracy over all test batches. Additionally, a confusion matrix was generated to provide a detailed breakdown of the model's performance across different classes, helping to identify any specific classes where the model's performance was suboptimal. An interactive prediction interface was also developed to allow users to upload images and receive predictions. Users could upload images of currency notes through a graphical user interface implemented using ipywidgets in a Google Collab Notebook environment. Uploaded images were preprocessed using the same pipeline applied to the training data (resizing and normalization). The preprocessed image was then fed into the trained CNN model, and the predicted class along with the confidence score was displayed, with the prediction results presented alongside the uploaded image for easy verification (Fig. 2.).

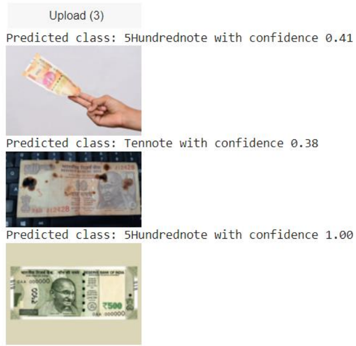



Fig. 2. Currency classification with confidence metric

## V. EXPERIMENTS AND RESULTS

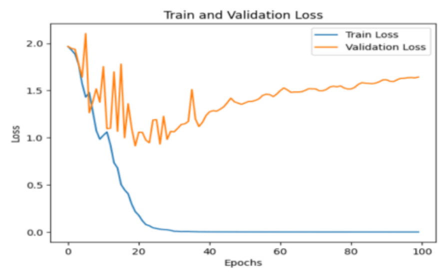
The performance of the proposed convolutional neural network (CNN) for Indian currency note classification was evaluated through several metrics and visualizations. The evaluation included accuracy measurement on the test set, loss and accuracy curves for both training and validation sets, a confusion matrix, and a detailed classification report.

### A. *Accuracy Measurement*

The final test accuracy achieved by the model was approximately 65.94%. This indicates that the model correctly classified about 65.94% of the test images.

### B. *Loss and Accuracy Curves*

The training and validation loss curves (Figure 2) showed a significant decrease in training loss over the epochs, eventually reaching near-zero values. However, the validation loss demonstrated fluctuations and a slight increasing trend, indicating potential overfitting. Similarly, the training and validation accuracy curves (Fig. 3.) illustrated that training accuracy reached 100% while validation accuracy plateaued around 69%, further suggesting overfitting.



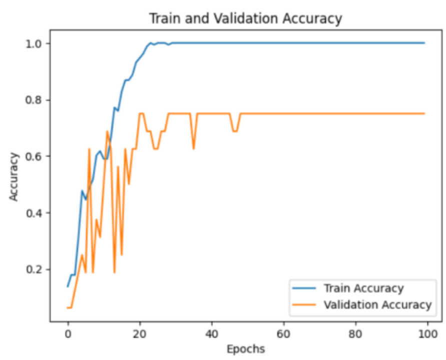



Fig. 3. Learning Curves for Loss and Accuracy

### *C. Confusion Matrix*

The confusion matrix (Fig. 4.) provided a detailed view of the model's performance across different classes. The matrix indicated that certain classes, such as the 5Hundrednote and Twentynote, were classified with higher accuracy, while others, like the Fiftynote and Tennote, experienced more misclassifications. The confusion matrix highlighted the specific areas where the model struggled to differentiate between similar-looking denominations.

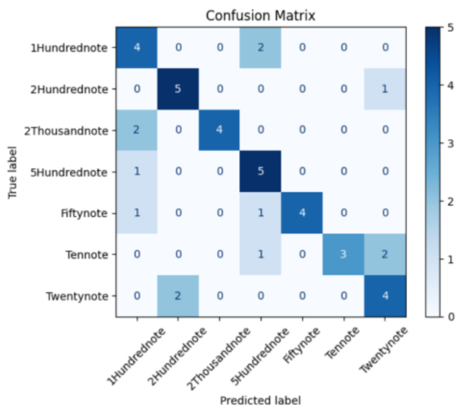



Fig. 4. Confusion matrix for model performance

### *D. Classification Report*

The classification report (Fig. 5.) detailed precision, recall, and F1-scores for each class. The precision ranged from 50% for the 1Hundrednote to 100% for the Fiftynote and Tennote. Recall values varied, with the 2Hundrednote achieving the highest recall at 83.33%. The F1-scores reflected these variations, indicating that certain classes were more accurately and consistently classified than others. The macro average and weighted average F1-scores were approximately 69%, providing a holistic view of the model's performance across all classes.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1Hundrednote | 0.500000 | 0.666667 | 0.571429 | 6.000000 |
| 2Hundrednote | 0.714286 | 0.833333 | 0.769231 | 6.000000 |
| 2Thousandnote | 1.000000 | 0.666667 | 0.800000 | 6.000000 |
| 5Hundrednote | 0.555556 | 0.833333 | 0.666667 | 6.000000 |
| Fiftynote | 1.000000 | 0.666667 | 0.800000 | 6.000000 |
| Tennote | 1.000000 | 0.500000 | 0.666667 | 6.000000 |
| Twentynote | 0.571429 | 0.666667 | 0.615385 | 6.000000 |
| accuracy | 0.690476 | 0.690476 | 0.690476 | 0.690476 |
| macro avg | 0.763039 | 0.690476 | 0.698482 | 42.000000 |
| weighted avg | 0.763039 | 0.690476 | 0.698482 | 42.000000 |

Fig. 5. Classification report

### *E. Discussion*

The results indicated that while the CNN achieved a reasonable level of accuracy in classifying Indian currency notes, there were several areas for improvement. The overfitting observed in the training and validation loss curves suggests that the model might benefit from regularization techniques such as dropout or weight decay. Additionally, data augmentation could be further enhanced to provide a more varied training set, potentially improving generalization.

The confusion matrix revealed specific classes where the model's performance was suboptimal. For instance, misclassifications between the 1Hundrednote and 2Thousandnote suggest that the model may be struggling with notes that have similar visual features or patterns. Further investigation into these misclassifications could provide insights into additional features or preprocessing steps that might aid in distinguishing these notes more effectively.

The classification report highlighted the variability in precision, recall, and F1-scores across different classes. While certain classes were classified with high precision and recall, others showed significant room for improvement. This variability underscores the importance of balanced datasets and the need for additional training data for underrepresented or more challenging classes.

## VI. CONCLUSION

In this study, a deep learning-based approach for the classification of Indian currency notes was presented, leveraging a convolutional neural network (CNN) architecture. The model was trained and evaluated using a dataset sourced from Kaggle, which included images of various denominations introduced both before and after the demonetization event in India. The CNN architecture, comprising multiple convolutional and max-pooling layers followed by fully connected layers, was designed to automatically learn hierarchical features from raw image data, enhancing the model's ability to generalize across different conditions and variations.

The experimental results demonstrated that the proposed CNN achieved a reasonable level of accuracy, with a test accuracy of approximately 65.94%. The training process, as visualized through the loss and accuracy curves, indicated potential overfitting, suggesting that the model may benefit from additional regularization techniques and enhanced data augmentation strategies. The confusion matrix and classification report provided detailed insights into the model's performance across different classes, highlighting areas where misclassifications occurred and underscoring the variability in precision, recall, and F1-scores.

Despite achieving significant improvements over traditional currency recognition methods, several challenges were identified. The observed overfitting and variability in classification performance across different denominations suggest that further enhancements are needed. Future work should focus on incorporating more sophisticated regularization methods, increasing the diversity and size of the training dataset, and exploring additional features that could aid in distinguishing visually similar notes.

The development of a robust and accurate currency recognition system has significant implications for financial institutions, particularly in the context of reducing the risk of counterfeit transactions and streamlining cash handling processes. The findings of this study contribute to the ongoing efforts to leverage deep learning technologies for practical applications in currency recognition and provide a foundation for future research aimed at achieving higher levels of accuracy and generalization.

The potential of deep learning in currency recognition has been demonstrated, yet the journey towards a fully reliable and efficient system continues. The integration of advanced machine learning techniques and the continuous improvement of model architectures will be crucial in overcoming the existing challenges and enhancing the capabilities of currency recognition systems in the real world.

## VII. PROJECT PROTOTYPE

The implementation details and source code for the currency recognition system described in this study are available on GitHub. The repository includes the dataset preprocessing scripts, the custom CNN architecture, training procedures, evaluation metrics, and the interactive prediction interface. Researchers and practitioners can access the complete project at the following link: https://github.com/ajaysasane8/Projects

This repository serves as a comprehensive resource for replicating and further enhancing the currency recognition system presented in this research.

## VIII. REFERENCES


[1] Brookings (2017) 'Early Lessons from India's Demonetization Experiment', Brookings. Available at: https://www.brookings.edu/blog/up-front/2017/01/31/early-lessons-from-indias-demonetization-experiment/

[2] LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436-444. Available at: https://www.nature.com/articles/nature14539

[3] Reserve Bank of India (2017) 'Macroeconomic Impact of Demonetisation', Reserve Bank of India. Available at: https://www.rbi.org.in/scripts/OccasionalPublications.aspx?head=Macroeconomic%20Impact%20of%20Demonetisation (Accessed: 23 May 2024).

[4] Jetir (2023) 'Impact of Demonetization on Indian Economy', Journal of Emerging Technologies and Innovative Research. Available at: https://www.jetir.org/papers/JETIR2304602.pdf

[5] Sahani, G. (2020) 'Indian Currency Notes Classifier', Kaggle. Available at: https://www.kaggle.com/datasets/gauravsahani/indian-currency-notes-classifier/data

[6] Bhurke, C.; Sirdeshmukh, M.; Kanitkar, M.S. Currency recognition using image processing. Int. J. Innov. Res. Comput. Commun. Eng. 2015, 3, 4418–4422.

[7] Pham, T.D.; Park, Y.H.; Kwon, S.Y.; Park, K.R.; Jeong, D.S.; Yoon, S. Efficient banknote recognition based on selection of discriminative regions with one-dimensional visible-light line sensor. Sensors 2016, 16, 328. Available at: https://www.mdpi.com/1424-8220/16/3/328