IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

# Deep Learning-Based Counterfeit Indian Currency Detection: A Comparative Study of ReLU and Leaky ReLU Activation Functions

**Ms. Devisha Sinha**
Assistant Professor, Department of Computer Science & Data Science
Nirmala Memorial Foundation College of Commerce and Science (Autonomous), Mumbai

**Abstract :** *Advancements in printing and scanning technologies have made it easier to produce counterfeit banknotes, which can reduce the value of real money and harm the economy. This paper examines the use of deep learning, particularly convolutional neural networks (CNNs), to detect counterfeit Indian currency notes from images. A CNN classifier is built that labels a note image as real or fake, and a comparison of two activation functions (Rectified Linear Unit, ReLU, and Leaky ReLU), two training lengths (20 and 100 epochs) and two optimisers (Adam and AdaGrad) is done under a common architecture. The images from a publicly available Kaggle dataset of Indian currency notes were resized to 224 × 224 pixels, normalised, and split into training, validation, and test sets in a 70:15:15 ratio. The Leaky ReLU model trained with Adam for 20 epochs reached 100% validation accuracy with a smooth learning curve, whereas the ReLU model trained for 100 epochs overfitted and reached only 80% validation accuracy. Since the evaluation subsets used in this draft are very small, the results are indicative rather than conclusive, and the data and experimental extensions needed to establish more definite findings are outlined.*

**Keywords:** Deep Learning, Convolutional Neural Network, Counterfeit Currency Detection, Rectified Linear Unit (ReLU), Leaky ReLU, Adam, Adaptive Gradient Algorithm (AdaGrad), Indian Currency.

## I. INTRODUCTION

### 1.1 Background

In recent years counterfeiting has threatened financial institutions, businesses and individuals worldwide. Fake notes reduce the purchasing power of genuine currency, undermine public trust in the monetary system and are often linked to wider illegal activity. As printers and scanners have become cheaper and more precise, counterfeit notes are difficult to tell apart from genuine ones just by sight, which makes automated detection increasingly important.

Most earlier detection systems depend on dedicated hardware and on hand-crafted image-processing pipelines. Typical steps include capturing the note, converting it to grayscale, detecting edges, segmenting regions, extracting features and comparing the result with a reference image. Variants use support vector machine (SVM) based recognition, segment the note into regions, or process images in RGB and HSV colour spaces. These techniques are effort-intensive and less effective when lighting, wear or noise change the appearance of a note.

Deep learning, a subfield of machine learning, has transformed feature extraction, image processing and pattern recognition. Instead of relying on hand-designed features, a deep network learns a hierarchy of features directly from images, offering a highly accurate, efficient and automated route to separating real from fake notes. Conventional neural networks have also been tried for this task, but they carry a higher risk of error and lower accuracy; convolutional neural networks are faster and more robust for image data and are therefore the most widely used choice.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

### 1.2 Problem Statement

Given an image of an Indian banknote, decide automatically whether the note is genuine or counterfeit. The decision should not depend on specialised scanning hardware, should generalise to unseen notes, and should be reached with a compact model that can be trained on modest resources. A recurring design question for such models is which activation function and optimiser give the most reliable learning behaviour, and this is the question the present study examines.

### 1.3 Motivation

ReLU is the default non-linearity in most deep networks. It is computationally cheap, mitigates the vanishing-gradient problem and produces sparse activations. Its weakness is that it outputs zero for every negative input, so neurons can stop updating altogether, a failure known as the "dying ReLU" problem. Leaky ReLU relaxes this by allowing a small non-zero slope for negative inputs. Whether this modification yields a practical benefit for banknote authentication, where the images contain fine texture and print patterns, is the motivation for the comparison carried out here.

### 1.4 Research Objectives and Questions

The objectives of this study are:

1. To design a CNN-based classifier that distinguishes real and counterfeit Indian currency notes from images.
2. To compare ReLU and Leaky ReLU activation functions in terms of accuracy, loss and learning behaviour.
3. To study the effect of training length (20 versus 100 epochs) and of the optimiser (Adam versus AdaGrad) on convergence and overfitting.
4. To identify the configuration that gives the most stable and accurate results and to document the limitations of the experiment.

These objectives lead to three research questions: (RQ1) Does Leaky ReLU give better accuracy and lower loss than ReLU on this task? (RQ2) How does the number of training epochs influence overfitting? (RQ3) How sensitive is the model to the choice of optimiser?

### 1.5 Scope of the Study

The study is deliberately bounded so that the comparison is controlled.

**In scope**

- Image-based binary classification (real versus fake) of Indian rupee notes using a publicly available Kaggle image dataset.
- A single custom CNN architecture, with the activation function (ReLU, Leaky ReLU), number of epochs (20, 100) and optimiser (Adam, AdaGrad) as the experimental variables.
- Data visualisation, pre-processing, training and evaluation on accuracy and loss, carried out in Google Colab using Keras/TensorFlow.

**Out of scope**

- Hardware-based or sensor-based verification (ultraviolet, magnetic, infrared, watermark or thread inspection).
- Currencies other than the Indian rupee, and detection of specific security features at feature level.
- Real-time or mobile deployment, adversarial robustness and explainability, which are discussed only as future work.

### 1.6 Contributions and Organisation of the Paper

The main contributions are:

1. a reproducible CNN pipeline for counterfeit Indian currency detection,
2. a controlled comparison of ReLU and Leaky ReLU across training lengths and optimisers, and
3. an explicit discussion of the limits of the current evidence.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

The rest of the paper is organised as follows. Section 2 reviews related work and states the research gap, while Section 3 provides the theoretical background. Section 4 describes the methodology and Section 5 presents and discusses the results. Finally, Section 6 lists the limitations, and Section 7 concludes with directions for future work.

## II. LITERATURE REVIEW

### 2.1 Image-Processing Approaches

Early counterfeit-detection systems combined hardware with classical image processing. A typical pipeline captures an image of the note, converts it to grayscale, detects edges, segments the image, extracts features and compares them with a genuine template. SVM-based currency recognition, region-wise segmentation of the note and processing in RGB and HSV colour spaces are common variants. Such methods depend on manually chosen features and on controlled acquisition conditions, which limits accuracy and makes them slow when many notes must be checked.

### 2.2 Neural-Network and Deep-Learning Approaches

Conventional neural networks were later applied to the problem, but they showed a higher risk of error and lower accuracy than modern convolutional models. Jadhav et al. [3] used the AlexNet deep-learning model for currency identification and forged-note detection in 2019. Because the network learns features layer by layer, the approach gives reliable detection from the currency image alone. The authors point to several directions for improvement: incorporating all the security features of the currency through structural design and extensive training data, handling noise present in captured images as a processing step, and using surface patterns of the note as features for recognition and fake-currency detection.

Subsequent work continues along this line. Deep-learning approaches to counterfeit detection [4], [9], currency detection and recognition [5], image-processing combined with deep learning [6], comparative studies of machine-learning paradigms [7] and deep-learning classification and condition assessment of Indian notes [8] all indicate that learned features are preferred over hand-crafted ones. Table 1 summarises the studies considered.

TABLE 1. Summary of related work reviewed

| Ref. | Year | Focus of the work (as indicated by the title) |
|---|---|---|
| [3] | 2019 | Currency identification and forged-banknote detection using deep learning (AlexNet-based) |
| [4] | 2020–21 | Counterfeit detection through a deep-learning approach |
| [5] | 2018 | Currency detection and recognition based on deep learning |
| [6] | 2024 | Identification of counterfeit currency using image processing and deep learning |
| [7] | 2024 | Comparative study of machine-learning paradigms for counterfeit currency detection |
| [8] | 2025 | Deep-learning model for automated classification and condition assessment of Indian currency notes |
| [9] | 2019 | Counterfeit currency detection using a deep convolutional neural network |

### 2.3 Activation Functions and Optimisers

Maas et al. [2] introduced the leaky rectifier as a modification of ReLU that keeps a small gradient for negative inputs, which reduces the number of inactive units. Kingma and Ba [1] proposed Adam, a first-order optimiser that combines momentum with RMSProp-style adaptive learning rates and has become a standard choice for training deep networks. These two components are the main variables examined in the present study.

### 2.4 Research Gap

The reviewed literature shows that deep learning is effective for currency recognition, but several gaps remain:

- **Hardware and hand-crafted dependence**. Many deployed or earlier systems still depend on specialised hardware or manually engineered features, which limits scalability.
- **Limited comparison of design choices**. Prior studies mostly report one network and one training

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

configuration. Little controlled evidence was found on how the choice between ReLU and Leaky ReLU, the number of epochs, and the optimiser affect accuracy, loss, and overfitting for banknote images.

- **Insufficient attention to overfitting.** Small banknote datasets make it easy for a network to memorise the training images; the relation between training length and generalisation is rarely examined explicitly.
- **Indian-currency focus.** Compared with other currencies, fewer studies address Indian notes of several denominations, and the future-work suggestions in the literature (noise handling, surface-pattern features, richer training data) remain largely open.

This study addresses the second and third gaps by running a controlled experiment on Indian rupee notes in which only the activation function, epochs and optimiser change, and by reporting the learning curves together with a frank assessment of how far the evidence goes.

## III. THEORETICAL BACKGROUND

### 3.1 Convolutional Neural Networks

A CNN processes an image through stacked convolutional layers that apply small learnable filters, pooling layers that reduce spatial resolution, and fully connected layers that map the extracted features to a class probability. Weight sharing across the image makes CNNs efficient and gives them the ability to detect local patterns such as edges, textures and printing artefacts wherever they appear in the note.

### 3.2 Rectified Linear Unit (ReLU)

ReLU is a non-linear activation function that introduces non-linearity into the network and alleviates the vanishing-gradient problem. It is computationally efficient and speeds up the training of neural networks. It is defined as
$f(x) = max(0, x)$.

### 3.3 Leaky ReLU

Leaky ReLU modifies ReLU by granting a slight, non-zero slope α to negative inputs instead of assigning them zero:
$f(x) = x \text{ for } x \geq 0, \text{ and } f(x) = \alpha x \text{ for } x < 0.$
Because negative inputs still pass a small gradient, neurons are less likely to become permanently inactive, which reduces the “dying ReLU” issue.

### 3.4 Optimisers

- **Adam:** can be viewed as a combination of RMSProp and stochastic gradient descent with momentum. It scales the learning rate using the squared gradients, as RMSProp does, and uses a moving average of the gradient instead of the raw gradient, as momentum does [1].
- **RMS Prop:** an improvement on AdaGrad in which the denominator is an exponentially decaying average, so the effective learning rate is not driven monotonically towards zero.
- **Ada Grad:** an adaptive-gradient method that scales each parameter's learning rate by the accumulated squared gradients; it works well for sparse data but its learning rate can shrink too quickly.

### 3.5 Loss Function

As the task is binary (real versus fake) and the output layer is a single sigmoid unit, the models are trained with binary cross-entropy, $L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$, where y is the true label and $\hat{y}$ the predicted probability of the note being real.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

## IV. METHODOLOGY

### 4.1 Research Design and Workflow

The study follows an experimental design: one CNN architecture is trained repeatedly while a single factor at a time (activation function, epochs, optimiser) is changed, and the learning behaviour of each variant is recorded and compared. The workflow, shown in Fig. 1, consists of importing libraries, loading the data from Google Drive into Colab, data visualisation, data pre-processing, model building, and testing accuracy and loss.

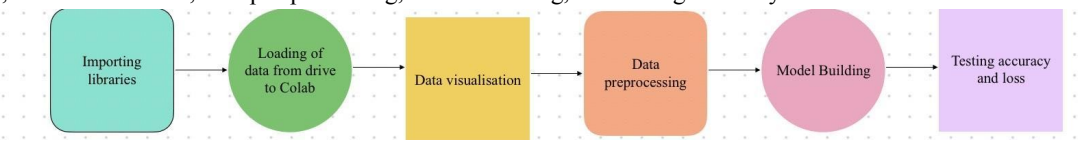



**Fig. 1. Flow chart of the proposed model.**

### 4.2 Dataset

The dataset was obtained from Kaggle and contains images of Indian rupee notes of several denominations, organised into separate training and testing folders. In our run the generator found 153 training images and 42 test images, organised in seven class folders. Representative samples are shown in Fig. 2.




Fig. 2. Sample of images in the dataset.

### 4.3 Data Pre-processing

Images are loaded with Keras' ImageDataGenerator, resized to 224 × 224 pixels with three colour channels and rescaled by 1/255 so that pixel values lie in [0, 1]. A batch of 32 images is drawn from the generator and split into training, validation and test subsets in the ratio 70:15:15 (22, 5 and 5 images respectively) using scikit-learn's train_test_split with a fixed random seed (42) for reproducibility.

### 4.4 Exploratory Data Visualisation

Before modelling, several visualisations were produced to understand the data: a bar graph of the class distribution, random image samples, augmented versions of an image, a histogram of pixel intensities for real and fake notes, and pair-wise image comparison. These are discussed in Section 5.1.

### 4.5 Model Architecture

The classifier is a sequential CNN with three convolution–pooling blocks followed by a dense layer and a sigmoid output (Table 2). The same architecture is used for every experiment; only the activation function in the convolutional and dense layers changes between the ReLU and Leaky ReLU variants.

**TABLE 2. CNN architecture used in all experiments**

| Layer | Configuration | Activation |
|---|---|---|
| Conv2D + MaxPooling2D | 32 filters, 3 × 3; pool 2 × 2 (input 224 × 224 × 3) | ReLU / Leaky ReLU |
| Conv2D + MaxPooling2D | 64 filters, 3 × 3; pool 2 × 2 | ReLU / Leaky ReLU |
| Conv2D + MaxPooling2D | 128 filters, 3 × 3; pool 2 × 2 | ReLU / Leaky ReLU |
| Flatten | – | – |

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

| Layer | Configuration | Activation |
|---|---|---|
| Dense | 64 units | ReLU / Leaky ReLU |
| Dense (output) | 1 unit | Sigmoid |

### 4.6 Training Configuration

Table 3 lists the configurations that were trained. All models use binary cross-entropy loss, a learning rate of 0.001 and a batch size of 32.

TABLE 3. Experimental settings

| Parameter | Value |
|---|---|
| Loss function | Binary cross-entropy |
| Learning rate / batch size | 0.001 / 32 |
| Activation functions compared | ReLU, Leaky ReLU |
| Epochs compared | 20, 100 |
| Optimisers compared | Adam (main experiments), AdaGrad (20-epoch comparison) |
| Data split (train : validation : test) | 70:15:15 |
| Framework and platform | Python, Keras / TensorFlow, Google Colab |

### 4.7 Evaluation Metrics

Models are assessed using training and validation accuracy and loss recorded after every epoch, which exposes overfitting through the gap between the curves. Accuracy, precision, and recall are additionally computed on the test subset. Precision measures how many notes predicted as real are truly real; recall measures how many of the real notes are found.

### 4.8 Experimental Variants

- ReLU with Adam, 100 epochs and 20 epochs.
- Leaky ReLU with Adam, 100 epochs and 20 epochs.
- The same CNN trained with the AdaGrad optimiser for 20 epochs, as an optimiser comparison.

## V. RESULTS AND DISCUSSION

### 5.1 Data Exploration

The class distribution of a sample batch of real and fake notes is shown in Fig. 3. The pixel-intensity histogram (Fig. 4) indicates that the dataset is relatively evenly distributed, with a good mix of dark, light and mid-tone pixels and no major outliers, which means that the images do not contain large uniformly dark or light areas.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

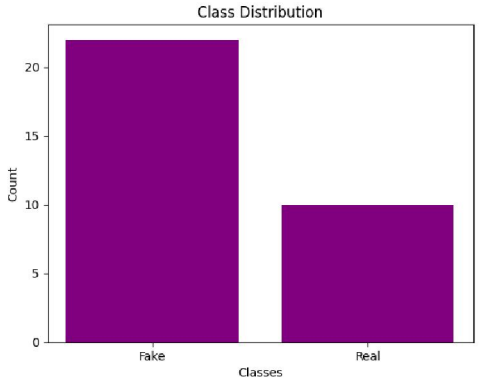



Fig. 3. Division of real and fake notes in a sample batch.

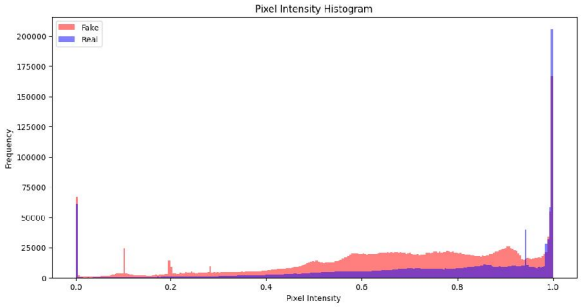



Fig. 4. Histogram of pixel-intensity frequency across the images in the dataset.

**5.2 ReLU Model**

With 100 epochs (Fig. 5), the training accuracy rises steadily, whereas the validation accuracy stays relatively flat and ends at 0.80 with a validation loss of 0.213. This gap shows that the model is overfitting: it learns patterns of the training images that do not generalise to new data (the training loss falls to almost zero). The training and validation loss both decrease over the run, while training accuracy increases. With 20 epochs (Fig. 6) the same architecture reached a validation accuracy of 1.00 and a validation loss of 0.0024.




Fig. 5. (a) ReLU, 100 epochs: training vs. validation accuracy

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

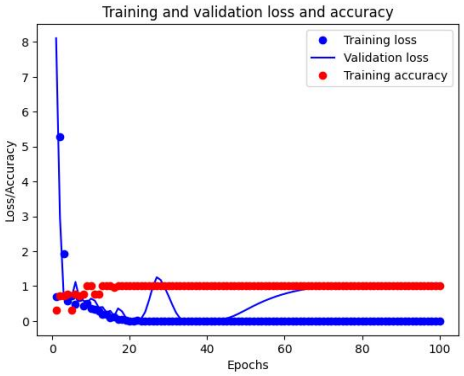



Fig. 5. (b) ReLU, 100 epochs: training and validation loss and training accuracy.

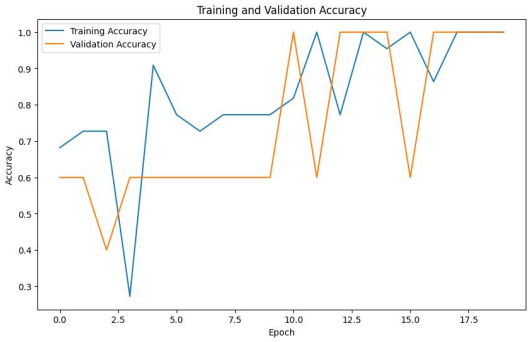



Fig. 6. (a) ReLU, 20 epochs: training vs. validation accuracy

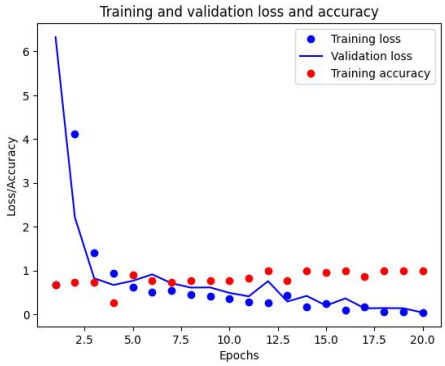



Fig. 6. (b) ReLU, 20 epochs: training and validation loss and training accuracy.

### 5.3 Leaky ReLU Model

The Leaky ReLU model trained for 100 epochs (Fig. 7) reaches a validation accuracy of 1.00 and then stagnates at this value after about the 28th epoch, so the remaining epochs add training time without improving the validation result. With Adam and 20 epochs (Fig. 8) the model learns more gradually: the final training accuracy is 0.95 with a training loss of 0.25, and the validation accuracy is 1.00 with a validation loss of 0.433, so the curves stay close to each other rather than separating.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-41170

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2




Fig. 7. (a) Leaky ReLU, 100 epochs: training vs. validation accuracy

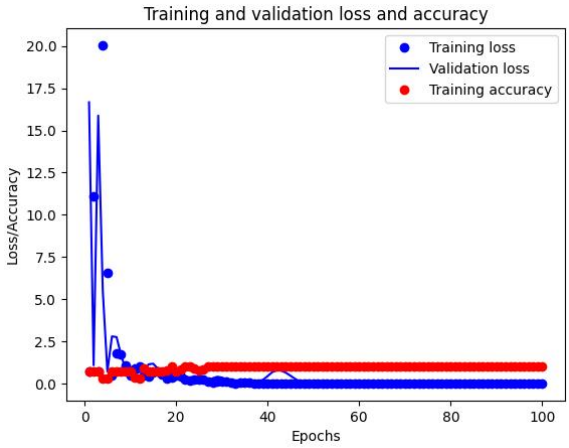



Fig. 7. (b) Leaky ReLU, 100 epochs: training and validation loss and training accuracy

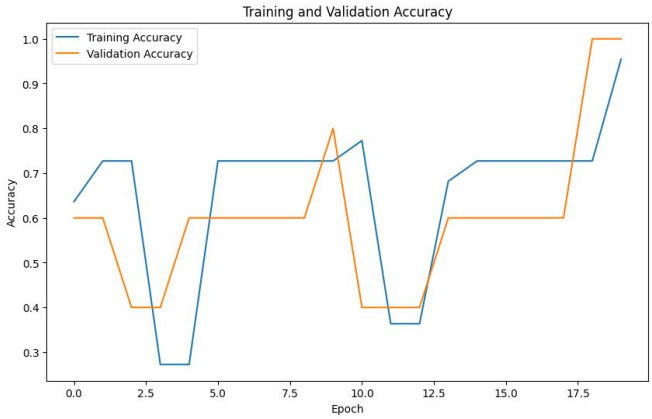



Fig. 8. (a) Leaky ReLU with Adam, 20 epochs: training vs. validation accuracy

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-41170

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

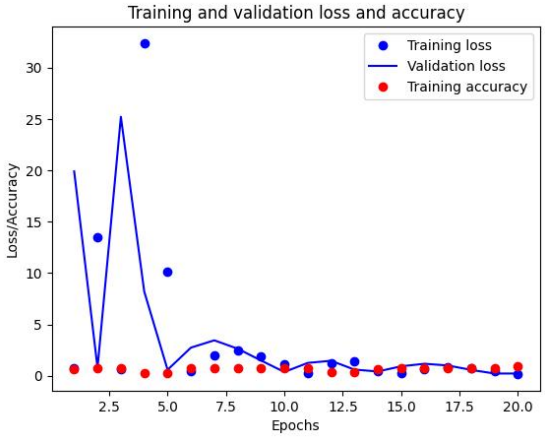



Fig. 8. (b) Leaky ReLU with Adam, 20 epochs: training and validation loss and training accuracy

### 5.4 Optimiser Comparison

Replacing Adam by AdaGrad (learning rate 0.001, 20 epochs) gave a clearly weaker result: the validation accuracy was 0.60 and the validation loss 0.715, and the training loss barely moved (about 0.56 in the epochs shown in the log). Adam is therefore retained as the optimiser for the main experiments.

### 5.5 Comparative Analysis

Fig. 9 compares the validation accuracy of the four activation/epoch variants, and Table 4 collects the numerical results recorded in the experiments.

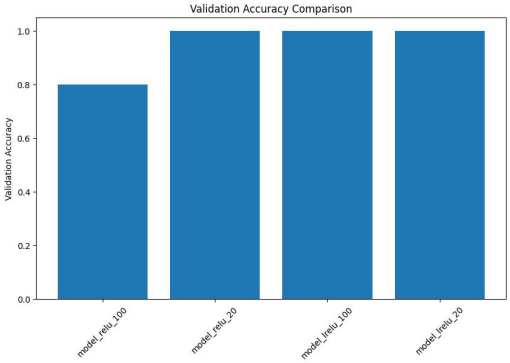



Fig. 9. Validation accuracy comparison for ReLU (100 and 20 epochs) and Leaky ReLU (100 and 20 epochs)

TABLE 4. Results of the experimental variants

| Model | Optimiser | Epochs | Val. acc. | Val. loss | Test acc. | Test prec. | Test recall |
|---|---|---|---|---|---|---|---|
| ReLU | Adam | 100 | 0.80 | 0.213 | 0.80 | 1.00 | 0.50 |
| ReLU | Adam | 20 | 1.00 | 0.0024 | 0.80 | 1.00 | 0.50 |
| Leaky ReLU | Adam | 100 | 1.00 | n/r | n/r | n/r | n/r |
| Leaky ReLU | Adam | 20 | 1.00 | 0.433 | n/r | n/r | n/r |
| ReLU | AdaGrad | 20 | 0.60 | 0.715 | n/r | n/r | n/r |

n/r = not recorded in the experiment log. Validation and test subsets contain only five images each, so all figures should be read as indicative.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-41170

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

### 5.6 Discussion and Model Selection

Leaky ReLU with Adam and 20 epochs was selected as the final configuration, for three reasons visible in the learning curves. First, it reached a validation accuracy of 1.00 without the sharp divergence between training and validation behaviour that characterised the 100-epoch ReLU model. Second, the 100-epoch Leaky ReLU model offered no further validation gain after about the 28th epoch, so the longer run only costs additional training time and increases the risk of overfitting. Third, the Leaky ReLU curves were smoother, consistent with the expectation that a non-zero negative slope keeps more neurons active during training.

In answer to the research questions: (RQ1) Leaky ReLU matched the best validation accuracy of 1.00 and learned more gradually, but on the present validation subset the ReLU model with 20 epochs also reached 1.00 (and a lower validation loss), so no decisive accuracy advantage of either activation function can yet be claimed. (RQ2) Longer training clearly increased overfitting for ReLU (0.80 validation accuracy at 100 epochs versus 1.00 at 20 epochs). (RQ3) Adam was markedly better than AdaGrad under identical settings (1.00 versus 0.60 validation accuracy).

## VI. LIMITATIONS

- Very small evaluation subsets. Because the 70:15:15 split was applied to a single batch of 32 images, the validation and test subsets hold five images each. One image changes accuracy by 20 percentage points, so a score of 1.00 is not strong evidence of generalisation.
- Limited dataset. The full dataset has 195 images (153 training, 42 test); this is small for deep learning and may not cover the range of lighting, wear and counterfeit quality seen in practice.
- Single runs. Each configuration was trained once, without repeated runs, cross-validation or confidence intervals, so run-to-run variance is unknown.
- Incomplete metrics. Precision and recall were recorded for the ReLU models only (recall of 0.50 shows that half of the real test notes were missed), and no confusion matrix was produced.
- No external validation. The model has not been tested on notes captured with different cameras or under different conditions.

## VII. CONCLUSION AND FUTURE WORK

### 7.1 Conclusion

This paper examined deep-learning-based detection of counterfeit Indian currency using a compact CNN and compared ReLU and Leaky ReLU activations, 20 versus 100 training epochs and the Adam and AdaGrad optimisers. The Leaky ReLU model trained with Adam for 20 epochs gave a validation accuracy of 100% with a smooth, non-diverging learning curve and was chosen as the preferred configuration, while the ReLU model trained for 100 epochs overfitted and AdaGrad performed worse than Adam. The results show that CNNs are a promising, hardware-independent way to authenticate banknotes. At the same time, the small evaluation subsets mean that the superiority of Leaky ReLU over ReLU is not yet statistically established.

### 7.2 Future Work

- Evaluate on the full test set with k-fold cross-validation, repeated runs and a confusion matrix, reporting precision, recall and F1-score.
- Enlarge and diversify the dataset (more notes, denominations, devices and lighting conditions) and use data augmentation to reduce overfitting; apply L2 regularisation, dropout and early stopping.
- Use transfer learning with pre-trained architectures and explore Siamese networks for comparing a note against a genuine reference.
- Incorporate specific security features (watermark, security thread, micro-lettering) and surface-pattern features, and make the model robust to image noise.
- Combine image analysis with sensor technologies, and pursue real-time processing, explainability and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

robustness to adversarial attacks, with a view to deployment on mobile and embedded devices in collaboration with the authorities.

## REFERENCES

[1] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations (ICLR), 2015.
[2] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," Proc. 30th International Conference on Machine Learning, 2013.
[3] M. Jadhav, Y. K. Sharma, and G. M. Bhandari, "Currency Identification and Forged Banknote Detection using Deep Learning," International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET), 2019, doi: 10.1109/ICITAET47105.2019.9170225.
[4] D. Saire and S. Tabbone, "Counterfeit Detection Through a Deep Learning Approach," 25th International Conference on Pattern Recognition (ICPR), 2020/2021, doi: 10.1109/ICPR48806.2021.9412224.
[5] "Currency Detection and Recognition Based on Deep Learning," 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, doi: 10.1109/AVSS.2018.8639124.
[6] "Identification of Counterfeit Currency Using Image Processing and Deep Learning," IEEE Conference Publication, 2024, doi: 10.1109/ICCCNT61001.2024.10724968.
[7] "Machine Learning Paradigms for Counterfeit Currency Detection: A Comprehensive Comparative Study," 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT), 2024, doi: 10.1109/AIIoT58432.2024.10574620.
[8] "Deep Learning Model for Automated Classification and Condition Assessment of Indian Currency Notes," 3rd International Conference on Disruptive Technologies (ICDT), 2025, doi: 10.1109/ICDT63985.2025.10986552.
[9] K. Kamble, A. Bhansali, P. Satalgaonkar, and S. Alagundgi, "Counterfeit Currency Detection using Deep Convolutional Neural Network," IEEE Pune Section International Conference, 2019.