IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13460

# Fake Currency Detection System

**Harshavardhan.G[1], Praveen.K[2], Bharath[3]**

CSE, SRM Institute of Science and Technology Chennai[1-3]

**Abstract:** This report covers the development and implementation of a Fake Currency Detection System that uses machine learning and image processing. The rise of counterfeit banknotes presents a serious threat to financial systems and public trust. This project tackles this issue by creating an automated software solution that accurately distinguishes genuine currency from counterfeit notes. The system uses Python, OpenCV, and Scikit-learn to process images of currency notes, extract meaningful features, and classify them with a Support Vector Machine (SVM) algorithm. A user-friendly Tkinter-based Graphical User Interface (GUI) allows users to upload an image and receive an instant prediction of authenticity. The project showcases how AI can improve financial security while providing a cost-effective and scalable alternative to traditional hardware detectors.

**Keywords:** Fake currency detection, machine learning, image processing, support vector machine, Tkinter GUI.

## I. INTRODUCTION

The global economy increasingly suffers from counterfeit currency circulation, which disrupts monetary stability, reduces consumer trust, and finances illegal activities.

Traditional detection methods often rely on costly specialized hardware, such as UV lights and magnetic sensors, or manual inspection. These methods can be slow and prone to human error. Integrating Artificial Intelligence (AI) and computer vision gives us an opportunity to automate and improve the accuracy of counterfeit detection.

*A. Problem Statement*

There is a pressing need for an accessible, accurate, automated system that can efficiently identify fake currency notes, especially for small businesses, banks, and retailers that may not have the budget for high-end detection equipment.

*B. Project Scope*

This project targets the development of a software solution for a specific currency denomination. The project includes:

- Dataset collection and preprocessing of currency note images.
- Feature extraction and model training using machine learning.
- Development of a standalone desktop application with a GUI.
- Evaluation and analysis of system performance.

## II. OBJECTIVES

The main and specific objectives of the Fake Currency Detection System project are outlined below.

*A. Primary Objective*

To design, develop, and deploy a reliable machine learning-based system that can automatically detect counterfeit currency notes from digital images.

*B. Specific Objectives*

- To gather and preprocess a labeled dataset with images of both genuine and fake currency notes.
- To apply image processing techniques for noise reduction, grayscale conversion, and normalization.
- To extract relevant numerical features from the preprocessed images for classification.
- To train, validate, and test a Support Vector Machine (SVM) classifier model.
- To achieve a target classification accuracy of over 92% on the test dataset.
- To create an easy-to-use desktop GUI application with Tkinter for user interaction.
- To comprehensively document the system architecture, methodology, results, and possible future improvements.

## III. LITERATURE SURVEY

A review of existing methodologies reveals several approaches to currency authentication, from hardware-based to software-driven techniques.

*A. Hardware-Based Methods*

Traditional detectors use ultraviolet (UV) light to reveal security threads, magnetic ink sensors, watermark detectors, and infrared scanners. While effective, these devices are dedicated units that carry high costs and limited flexibility.

*B. Image Processing & ML Approaches*

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Impact Factor 8.311 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 4, April 2026
DOI: 10.17148/IARJSET.2026.13460

Recent research emphasizes image analysis to extract features like texture, color histograms, and wavelet transforms. Machine learning models such as K-Nearest Neighbors (KNN), Random Forest, and Neural Networks have been used with varying success rates, often reaching over 90% accuracy in controlled environments.

*C. Identified Research Gap*

Many proposed systems still remain in the research phase and lack a practical, user-friendly interface for non-technical users. This project addresses that gap by combining a trained ML model with a fully functional desktop application, making AI-powered detection more accessible.

## IV. METHODOLOGY

The project follows a structured approach, dividing the work into phases that include data collection, model development, and application deployment.

Project Development Timeline
Phase
Description
Weeks
Requirement & Planning
Problem analysis, objective definition, tool selection
2
Data Acquisition & Prep
Dataset collection, preprocessing, feature extraction
3
Model Development
Algorithm selection, training, validation, testing
3
GUI & System Integration
Tkinter GUI development, model integration
2
Testing & Deployment
System testing, performance evaluation, final report
2

*A. Data Collection & Preprocessing*

A dataset of over 500 images of real and fake notes was compiled. Each image underwent preprocessing:

- Grayscale Conversion: Changing from RGB to a single channel to reduce complexity.
- Resizing: Adjusting all images to a fixed dimension (e.g., 256×256 pixels).
- Noise Reduction: Using Gaussian blur to lessen image artifacts.
- Normalization: Scaling pixel intensity values to a standard range.

*B. Feature Extraction & Model Training*

Preprocessed images were converted into 1D feature vectors. The dataset was divided into 70% for training, 15% for validation, and 15% for testing. A Support Vector Machine (SVM) with a linear kernel was trained on the feature vectors to classify notes as "Real" or "Fake."

## V. SYSTEM ARCHITECTURE

The system architecture is organized in a modular pipeline to ensure clarity, maintainability, and efficiency.

*A. Input Module*

Accepts a currency note image through the GUI. Supports common formats (JPG, PNG).

*B. Processing Engine*

The core component that uses OpenCV for grayscale conversion, resizing, and noise reduction.

*C. Feature Extractor*

Transforms the processed image into a numerical feature vector (flattened pixel array).

*D. ML Classifier*

The pre-trained SVM model processes the feature vector and calculates the prediction probability.

*E. Output Module*

Displays the result ("Genuine" or "Counterfeit") along with a confidence score on the GUI.

## VI. IMPLEMENTATION

The implementation phase involved coding the modules, integrating components, and creating the final application.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Impact Factor 8.311 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 13, Issue 4, April 2026
DOI: 10.17148/IARJSET.2026.13460

*A. Technical Stack*

• Programming Language: Python 3.8+

• Libraries: OpenCV (cv2) for image processing, NumPy for numerical operations, Scikit-learn for SVM modeling, Tkinter for GUI, Matplotlib for visualization.

• Environment: Jupyter Notebook for prototyping, PyCharm/VSCode for development.

*B. Key Implementation Steps*

• Loading the dataset and applying preprocessing functions to all images.

• Splitting data and training the SVM classifier using Scikit-learn's SVC class.

• Saving the trained model using Pickle for persistence and reuse.

• Designing the Tkinter GUI with buttons for uploading images and predicting results, along with labels for displaying outcomes.

• Writing the main application script to connect GUI events with the model prediction process.

Project Budget Overview
Item
Description
Cost
Development Tools
Software licenses (Open Source), cloud credits for training
50
Data Acquisition
Curating and labeling image dataset
200
Developer Hours
Approx. 120 hours at $25/hour
3,000
Testing & Deployment
Application packaging and distribution
100
Total Projected Cost
3,350

## VII. Results & Analysis

The system was thoroughly tested to evaluate its performance against the defined objectives. The trained SVM model was assessed on the unseen test dataset.

*A. Performance Metrics*

• Accuracy: 94.2%

• Precision: 93.5%

• Recall: 95.0%

• F1-Score: 94.2%

These metrics indicate a reliable classifier that meets the target accuracy goal.

*B. Confusion Matrix Analysis*

The confusion matrix showed a low rate of false positives (genuine notes misclassified as fake) and false negatives (fake notes misclassified as genuine), indicating balanced performance.

## VIII. Advantages

*A. Cost-Effectiveness*

Removes the need for costly dedicated hardware; works on standard computers.

*B. High Accuracy & Speed*

Offers near-instant results with over 94% accuracy, reducing human error.

*C. User-Friendly Interface*

The straightforward GUI makes the technology easy to use for individuals with no technical background.

*D. Scalability & Adaptability*

The software-based model can be updated or retrained easily for new currency designs or denominations.

## IX. Limitations

• Single Currency Focus: The model is trained for a specific currency denomination only.

DOI: 10.17148/IARJSET.2026.13460

• Image Quality Dependency: Performance may drop with poor lighting, blurriness, or extreme angles in the input image.
• Feature Simplicity: Using flattened pixel vectors may not capture complex security features as effectively as deep learning methods.
• Desktop-Only: The current solution is a desktop application, lacking mobile or web access.

## X. FUTURE SCOPE

The project sets a strong foundation for multiple impactful improvements and expansions.
Future Enhancement Roadmap
Enhancement
Description
Expected Benefit
Deep Learning Integration
Introduce Convolutional Neural Networks (CNNs) for automatic feature learning from raw images.
Boost accuracy to over 98% and improve robustness.
Multi-Currency Support
Train models on datasets featuring multiple global currencies and denominations.
Broaden market applicability and utility.
Real-Time Detection
Incorporate camera feeds for live analysis in ATMs or point-of-sale systems.
Allow for immediate detection in transactional settings.
Cloud & Mobile Deployment
Create a web API or mobile app for easy access and centralized model updates.
Significantly expand user access and convenience.

## XI. CONCLUSION

The Fake Currency Detection System project effectively shows that machine learning and image processing can address important financial security issues. By achieving a classification accuracy of 94.2% and packaging the model into a functional Tkinter GUI application, the project provides a practical tool for identifying counterfeits. The system supports the idea that software-based AI solutions can enhance traditional security measures. While the current version has limitations, it lays a solid groundwork for future development. The suggested improvements in deep learning, multi-currency support, and real-time deployment pave the way for turning this prototype into a commercially viable product, contributing significantly to the global fight against financial fraud.

## REFERENCES

[1] Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing.