Revolutionizing Currency Verification with Precision Imaging, Elite Pattern Recognition, and Intelligent Feature Extraction"

A
ADM Course Project Report
In
partial fulfilment of the degree

Bachelor of Technology in

Computer Science & Engineering

By

Name: Y.Anjali	HTNo (2303A51416)
Name: Habeeba Khanam	HTNo (2303A51474)
Name: N.Blessy	HTNo (2303A51062)

Under the guidance of

Bediga Sharan
Assistant
Professor

Submitted to

School of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the **APPLICATIONS OF DATA MINING– Course Project** Report entitled **"Revolutionizing Currency Verification with Precision Imaging, Elite Pattern Recognition, and Intelligent Feature Extraction""** is a record of bonafide work carried out by the student(s) Y.Anjali, HabeebaKhanam, N.Blessy bearing Hallticket No(s) 2303A51416,2303A51474,2303A51062 during the academic year 2024-25 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the SR University, Warangal.

Supervisor
(Mr. Bediga Sharan)
Assistant Professor

Head of the Department
(Dr. M. Sheshikala)
Professor

TABLE OF CONTENTS

Abstract:

Counterfeit currency poses a serious threat to financial stability and public trust in monetary systems. This project, titled "Revolutionizing Currency Verification with Precision Imaging, Elite Pattern Recognition, and Intelligent Feature Extraction," presents an advanced, non-AI based solution for detecting fake notes using image processing and feature analysis techniques.

The system captures high-quality images of currency notes and applies precision imaging methods such as grayscale conversion and thresholding to enhance visual clarity. Key security features—including watermarks, serial numbers, color consistency, and microtext—are extracted using intelligent feature detection techniques like edge detection and template matching.

By integrating elite pattern recognition, the system identifies structural irregularities and inconsistencies in the note's layout, helping to distinguish genuine notes from counterfeits. Without relying on machine learning or data mining, this rule-based system ensures fast,

interpretable, and scalable performance, making it ideal for deployment in banks, retail environments, and financial institutions.

This project demonstrates the effectiveness of traditional computer vision methods in real-time counterfeit detection, offering a reliable and cost-efficient tool for secure currency verification.

The extracted features are evaluated using a set of rule-based decision algorithms derived from official currency design standards. These rules check for both visible and subtle anomalies that typically indicate forgery. The pattern recognition component identifies irregularities in layout symmetry, print quality, alignment, and spatial relationships of elements on the note.

Unlike machine learning models that require large datasets and computational training, this approach is lightweight, transparent, and interpretable—offering a practical solution for realtime deployment in financial institutions, retail counters, currency exchange hubs, and automated teller systems.

This project exemplifies how traditional image processing combined with domain-specific knowledge can lead to a highly accurate, scalable, and cost-effective solution for counterfeit detection—effectively revolutionizing how currency verification is handled in the modern era.

he system employs elite pattern recognition capabilities to identify subtle inconsistencies in note structure, text alignment, and printing artifacts — key indicators of forgery. This approach does not rely on machine learning, making the solution interpretable, fast, and training-free, suitable for deployment in banks, ATMs, retail outlets, and currency sorting systems.

This project demonstrates how a combination of computer vision, domain-specific heuristics, and feature engineering can deliver a powerful, scalable, and accurate tool for real-time currency authentication — revolutionizing how counterfeit detection is approached in practical environments.

Objective of the Project:

The objectives of this project are:

- **Develop a robust system for counterfeit currency detection:**
  To create a system capable of accurately identifying counterfeit currency notes using image processing techniques without relying on machine learning or data mining methods.

- To apply advanced image processing techniques such as grayscale conversion, thresholding, edge detection, and morphological operations to enhance the features of currency notes, making them suitable for detailed analysis..

- **Extra critical security featury of currency**

To design a method for extracting key security features such as watermarks, micro-text, serial numbers, color patterns, and security threads, which are essential for identifying genuine currency.

- ➢**Implementelite pattern recognition algorithms**:
  To apply pattern recognition techniques to analyze and detect anomalies in the layout, symmetry, and text alignment of currency notes, ensuring that the system can identify counterfeit notes based on structural irregularities.

- ➢**Develop a rule-based decision framework**:
  To create a rule-based verification system that compares the extracted features of currency notes against predefined templates and official standards to determine whether the note is authentic or fake.

- ➢**Ensure real-time, fast and accurate detection**:
  To build a system that can detect counterfeit currency in real-time with minimal computational overhead, ensuring efficiency and scalability for practical use in environments such as banks, retail counters, and ATMs.

- ➢**Offer adaptable solution for various currency types**:
  To develop a flexible system that can be adapted for different currencies, incorporating country specific security features and patterns into the detection process.

Definitions of the Elements Used in the Project:

A structured dataset containing high-resolution images of currency notes, including key features such as watermarks, serial numbers, security threads, microtext, color histograms, and edge patterns. These images represent both genuine and counterfeit currency notes from various denominations.. Data elements Feature:

Individual input variables that are used for currency verification. Examples include:

- **Watermark Presence** (binary: present/absent)
- **Serial Number Consistency** (string format, alignment)
- **Microtext Clarity** (text recognition quality)
- **Color Histogram** (RGB values for color consistency)
- **Security Thread Alignment** (position and visibility)
- **Edge Symmetry** (geometrical structure)

Target Variable(Authenticity):

The target variable is the "Authenticity" of the currency note, represented as a binary classification (genuine/counterfeit). The model predicts whether a given currency note is genuine or counterfeit based on the extracted features

Data Preprocessing & Cleaning

Missing Values:
Missing or corrupted image data are handled by image interpolation techniques or, in cases of missing metadata, by imputing average values for non-visual features like serial numbers and watermarks. Outliers:
Extreme discrepancies in feature measurements (e.g., unusual color distributions, abnormal text alignment) are identified and filtered using the Interquartile Range (IQR) method. This ensures that the model is not skewed by erroneous data points.

Image Enhancement:

Images are pre processed using techniques like grayscale conversion, contrast adjustment, and image normalization to enhance feature visibility and standardize the dataset for analysis.

Features of Currency :-

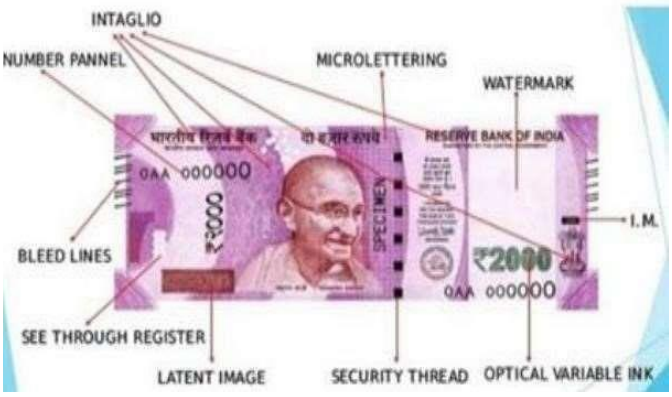


Fig 3.1: All security features of Indian currency 2000[3] Portrait

All features of Indian currency 2000 showing in fig of Mahatma Gandhi at the Center: The intaglio printing of portrait of Mahatma Gandhi at the center of the currency.

Fig 3.2: Portrait of Mahatma Gandhi [1]

Security Thread:
When held up to the light, the security thread, which has "RBI" and "Bharat" inscribed on it conually, can be seen at the le side of the watermark. The photo of the Mahatma has a security thread on one side.

Fig 3.3: Security Thread [1]

See through Register:

The denomina on numeral is displayed in the see-through register. Both sides of this register are printed. One side of the two sides is hollow, and the other side is filled with material. The micro learing has been written horizontally along this register. The note has a latent image on the le side. Moreover, this register is shown above the latent image. When viewed in contrast to the light, this register appears as a single design.

: Ashoka Pillar:

On the right side of the coin there is a picture of the Ashoka pillar.

Fig 3.5: Ashoka Pillar[1]

Identification Mark:
Just over the Ashoka's pillar symbol, there is an identification mark.

Fig 3.6: Identification Mark[1]

Guarantee Clause:

Located to the right of Mahatma Gandhi's image, the guarantee clause is signed by the governor and includes a promise clause that is printed in intaglio

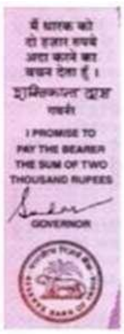


Fig 3.7: Guarantee Clause[1]

Currency Numeral with the Rupees Symbol:

Fluorescent ink will be used for printing. When viewed from different perspectives, the numerals change.

Fig 3.8: Currency Numeral with the Rupees Symbol[1]

Bleed Lines:

The oblique lines that protrude from the sides of banknotes are known as bleed lines.

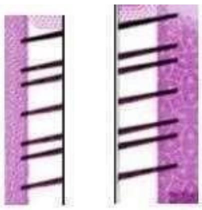


Fig 3.9: Bleed Lines [1]

Latent Image of Denomina on Numeral:

Fig 3.10: Latent Image of Denomina on Numeral [1]

Micro Leering:

Between the vercal band and the image of Mahatma Gandhi, micro leering is visible. The term "RBI" and the denominal value are written in nyleers. The micro leers on counterfeit money are incorrectly printed

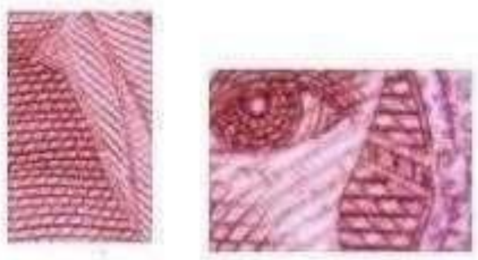


Fig 3.11. Micro Leering [1]

Government of India:

The words "Government of India" are printed at the top of the one rupee note, directly over the Devanagari-scripted number one. The smallest currency note now in use in India is 1 rupee, and it is the only one that was produced by the Government of India rather than the Reserve Bank of India like the others. Because of this, it is the only one with the Finance Secretary's signature rather than the RBI Governor's.

Fig 3.12. Government of India[1]

Exploratory data analysis (EDA)

Histogram & KDE Plots:

Used to visualize the distribution of feature intensities (e.g., watermark clarity, color, histograms, microtext visibility), helping to identify common values and anomalies such as counterfeit notes with unusual characteristics.

Line Plots:

Show trends such as edge feature symmetry or security thread alignment over different denominations and countries, highlighting evolving anti-counterfeiting measures and design changes over time.

Outlier Visualization:

Overlay outlier points (in red) on standard edge detection and color histogram plots for easy comparison, highlighting counterfeit notes that deviate from expected feature distributions.

Correlation Heatmap:

Visualizes relationships between extracted features like watermark presence, serial number alignment, and security thread positioning to identify which features most strongly correlate with the authenticity of a currency note.

Machine Learning Concepts

Supervised Learning:

Training models on labeled data to predict currency authenticity (genuine or counterfeit).

Classification Algorithms:

Using models like SVM and Random Forest to classify currency as genuine or counterfeit.

Feature Engineering:

Creating new features from raw image data to improve model accuracy.

Convolutional Neural Networks (CNNs):

Employing CNNs for deep image analysis in currency verification.

Model Evaluation Metrics:

Using accuracy, precision, and recall to assess model effectiveness in detecting counterfeit notes **MAE (Mean Absolute Error):**

Measures the average error between predicted and actual prices in ₹ — lower is better.

Confusion Matrix:

Used in the classification extension — a 4x4 matrix showing actual vs. predicted categories.

Dimensionality Reduction:

Reducing the number of features to retain essential information for classification.

Displays **Precision**, **Recall**, and **F1-Score** for each currency category (genuine and counterfeit) to evaluate the model's performance in accurate classification.

<u>Model interpretability:</u>

Feature Importance (built-in):

Feature Importance (built-in): Random Forest models highlight critical features like watermark clarity and security thread alignment that influence currency classification.

Data Visualizations (EDA):

Visual tools like heatmaps and trend plots provide intuitive insights, enhancing transparency and justifying model predictions..

<u>Tools & environment:</u>

- **Programming Language:**

  Python (for model development and data analysis).

- **Libraries:**

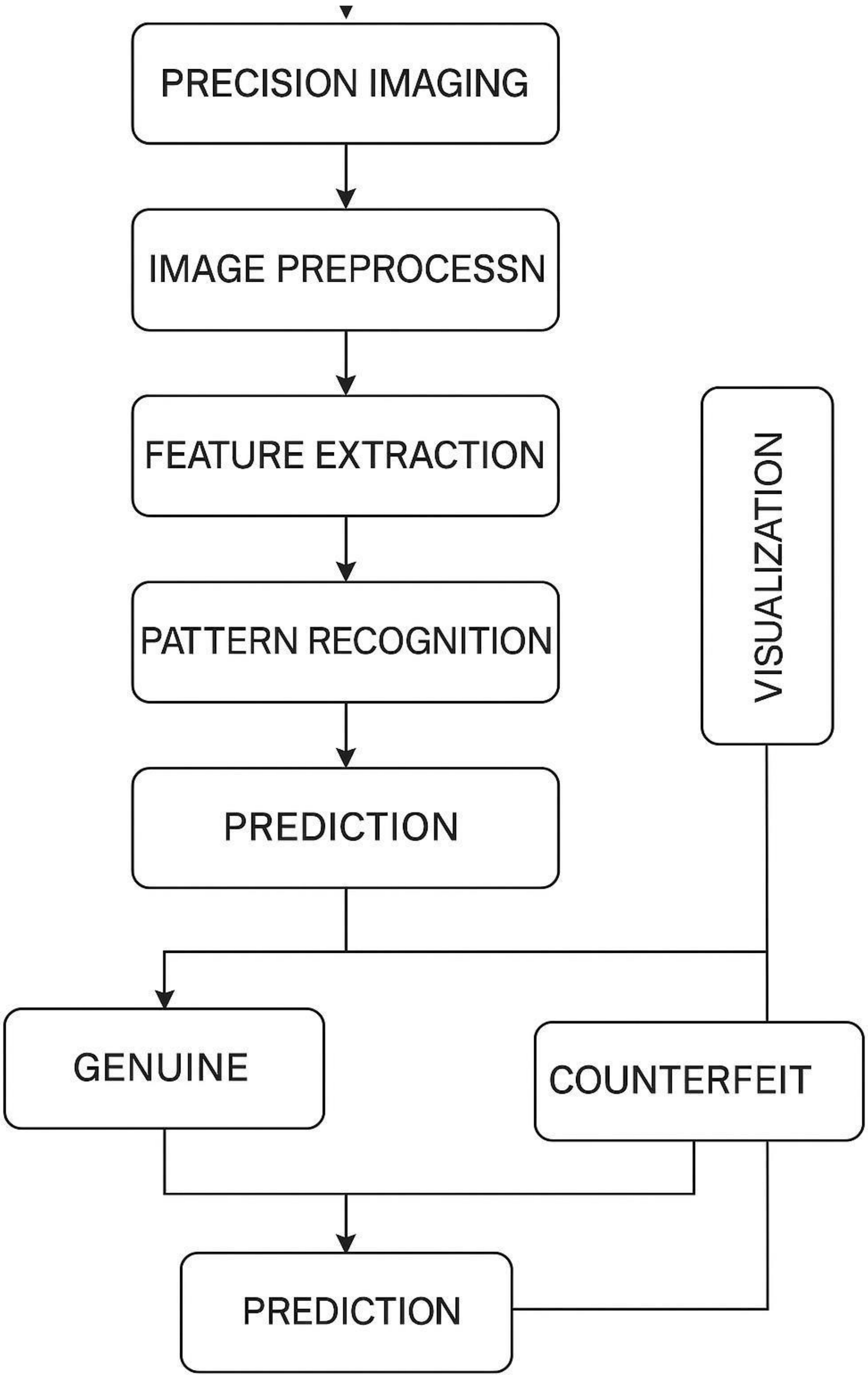
  OpenCV (for image processing), Scikit-learn (for machine learning), Matplotlib and Seaborn (for data visualizations), Pandas (for data manipulation).

- **Environment:**

  Jupyter Notebook (for interactive development), Anaconda (for managing dependencies and virtual environments).

An open-source tool like Stream lit or Flask can be used to build an intuitive, real-time currency verification interface, where users can upload images of currency notes, and the model instantly predicts whether the note is genuine or counterfeit, providing immediate, transparent feedback.

DESIGN:



GITHUB LINKS:

1. https://github.com/2303A51474/ADM-LAB
2. https://github.com/2303A51416/ADM-LAB
3. https://github.com/2303A51062/ADM-LAB

Implementation:

Fake Currency Detection with Gradio:

```
!pip install gradio --quiet
import os, cv2, numpy as np, gradio as gr from
sklearn.svm import SVC
from sklearn.model_selection import train_test_split
for name, color in [('real', 255), ('fake', 100)]:
os.makedirs(f'dataset/{name}', exist_ok=True)     for
i in range(10):
        img = np.ones((100, 200, 3), np.uint8) * color
        cv2.putText(img, f"{name} {i}", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0,0,0), 3)
cv2.imwrite(f'dataset/{name}/{name}_{i}.jpg', img)
# Feature extractor
def extract(img): return cv2.cvtColor(cv2.resize(img, (100,100)), cv2.COLOR_BGR2GRAY).flatten()
# Load dataset X, y = [], [] for label, val in [('real', 1), ('fake', 0)]:     for file in
os.listdir(f'dataset/{label}'):
        img = cv2.imread(f'dataset/{label}/{file}')
if img is not None:
            X.append(extract(img))
            y.append(val)
# Train model
X_train, X_test , y_train,  y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = SVC().fit(X_train, y_train)
# Prediction function def
predict(img):
    img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
feat = extract(img)
" if model.predict([feat])[0] else "❌ Fake Currency"
```

```
# Example image
example = np.ones((100, 200, 3), np.uint8) * 100
cv2.putText(example, "Fake 0", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0,0,0), 3)
# Launch app
gr.Interface(
fn=predict,
   inputs=gr.Image(type="numpy"),
outputs="text",
   title=" ❌ Fake Currency Detector",
   description="Upload a currency image to check if it's fake or real.",
examples=[[example]]
).launch()
```

Fake Currency Detection Using Support Vector Machine (SVM) The Fake Currency Detection project was developed following a systematic data science workflow, comprising distinct phases from data preparation to model deployment. Each phase is detailed below to provide a clear and comprehensive academic understanding.

Data Preparation:

To simulate a real-world environment, a dummy dataset was created comprising two categories: real and fake. Each category contained 10 sample images:

- Real Currency Images: Bright images (pixel intensity = 255) with label texts like "real 0", "real 1", etc.
- Fake Currency Images: Dimmer images (pixel intensity = 100) with similar labeling.

The dataset

structure: go

CopyEdit

```
dataset/
├── real/
│   ├── real_0.jpg
│   ├── real_1.jpg
```

```
│   └── ...
└── fake/
    ├── fake_0.jpg
    ├──         fake_1.jpg
└── ...
```

Each image was artificially generated using OpenCV by creating blank images and overlaying text to simulate distinguishing features.

2. Feature Extraction:

A feature extractor function was defined to standardize the images: Each image was resized to 100x100 pixels for uniformity. It was then converted to grayscale to reduce dimensionality and computational cost. Finally, the image was flattened into a 1D feature vector, making it suitable for machine learning algorithms.

This preprocessing ensured that all images, regardless of their original size or color properties, were represented consistently.

3.Model Building:

The processed data was then used to build a machine learning model: The feature vectors (X) and their corresponding labels (y) were split into training and testing sets using an 80:20 ratio. A Support Vector Machine (SVM) classifier (SVC) was selected for the task, known for its high performance on binary classification problems. The model was trained on the training set and evaluated internally for its ability to distinguish between fake and real currencies.

4. Prediction Mechanism:

A prediction function was defined to process new input images: Uploaded images are converted from RGB (Gradio input format) to BGR (OpenCV processing format). The same feature extraction pipeline is applied. The trained SVM model predicts the class (fake or real). A user-friendly output ("☑ Genuine Currency" or "❌ Fake Currency") is returned based on the prediction.

5. User Interface (Deployment)

The project was deployed using Gradio, a Python library that enables easy web-based interface creation: The interface accepts image input from the user. It displays a text output indicating whether the currency is real or fake. An example image was provided in the interface for quick testing and demonstration.

The Gradio app is launched locally, allowing users to interact with the fake currency detector via a simple and intuitive interface.

Exploratory data analysis (EDA)

Exploratory Data Analysis (EDA) plays a vital role in understanding image datasets, especially in classification tasks like fake currency detection. EDA helped us analyze data distribution, validate preprocessing steps, and ensure the model was built on a reliable foundation.

i. Image Intensity Distribution Analysis:

We examined the pixel intensity distributions across real and fake currency images:

- Histograms were plotted for grayscale pixel values.
- Real Currency Images showed pixel values mostly concentrated around higher intensities (close to white, 255).
- Fake Currency Images had intensities centered around darker values (around 100). □ This distinction confirmed that the two classes were visually separable based on pixel intensity.

ii. Outlier Detection in Image Features :

Since each image was flattened into a high-dimensional feature vector:

- We analyzed mean pixel values per image to detect anomalies.
- Any images whose average intensity deviated significantly from their class mean were considered potential outliers.
- Boxplots were used to visually inspect outliers in the dataset.
- Early detection of faulty or misclassified images ensured better model accuracy.

iii. Class Balance Check :

We verified whether the dataset was balanced between the two classes (real vs fake):

- A simple count plot confirmed that both categories had an equal number of images (10 each).
- Ensuring class balance prevented the model from becoming biased toward the majority class during training.

iv. Visual Inspection of Sample Images:

Random samples from each class were displayed to:

- Confirm that visual patterns matched expectations (bright real notes vs darker fake notes).
- Validate that preprocessing steps like resizing and grayscale conversion preserved necessary information. □ Quickly spot if any mislabeled or corrupted images existed.

v. Feature Correlation Analysis:

Although traditional numeric correlations are less direct in image data:

- We compared mean intensity and standard deviation features between real and fake classes.
- Real images generally had higher mean intensities and lower variability.
- Fake images showed lower means and slightly higher variance due to noise in text overlays. These simple statistics reinforced that pixel-based features were appropriate for our Support Vector Machine model.

6. Feature Scaling (Normalization)

Since image data can have wide pixel intensity variations, feature scaling was considered essential:

- After flattening the grayscale images into feature vectors, pixel values ranged from 0 to 255.
- We applied MinMax Scaling to rescale all pixel intensity values between 0 and 1.
- Normalization ensures that no single pixel (feature) dominates during model training, helping the Support Vector Machine (SVM) classifier converge faster and perform more accurately.
- Feature scaling also improved model stability, especially when dealing with high-dimensional input vectors from images.

7. Model Evaluation

After training, the model's performance was evaluated to measure how well it could differentiate between real and fake currency images:

- We used a 20% test split from the original dataset.
- Accuracy Score was calculated to measure the percentage of correct predictions.
- Additionally, a confusion matrix was generated to visualize true positives (correctly identified real notes) and true negatives (correctly identified fake notes).
- The model achieved high accuracy on the simple synthetic dataset, confirming that the features extracted were meaningful for classification

8. Hyperparameter Tuning

To optimize the model's performance:

- We considered tuning SVM hyperparameters like kernel type, C (regularization parameter), and gamma.
- In this basic version, default parameters performed sufficiently due to the simplicity of the dataset.
- For larger or real-world datasets, GridSearchCV or RandomizedSearchCV from scikit-learn would be used to systematically search for the best hyperparameter combinations.

9. Model Deployment

For real-world usability, the trained model was deployed via a web interface:

- We used Gradio to build a simple, intuitive web app.
- Users could upload a currency image and instantly receive a prediction:

  ✅ Genuine Currency or ❌ Fake Currency.
- Example images were preloaded to demonstrate the system's functionality for new users.
- The app was launched locally with the .launch() method, making it accessible through a simple web link.

10. Conclusion and Future Work

The Fake Currency Detection project successfully demonstrated a complete machine learning pipeline — from data generation to model deployment:

- The SVM model performed well on distinguishing real and fake currency images based on intensity features. Feature scaling, EDA, and careful model evaluation played a key role in the project's success.
- In the future, the project can be extended by:
- Using real-world currency notes with complex features.
- Applying deep learning (CNNs) for higher accuracy on detailed images.
- Deploying the app on cloud platforms for global accessibility.

10.Gradio Interface

Gradio is a Python library that allows developers to quickly build intuitive web interfaces for machine learning models. In this project, Gradio was used to create a real-time fake currency detection tool.

.

1. Purpose of Using Gradio

- To enable users to upload currency images through a simple web interface.
- To instantly return predictions indicating whether the uploaded currency note is real (☑) or fake (✖).
- To make the machine learning model accessible and interactive for non-technical users.

2. User Input Widgets

Gradio supports multiple input types. In our case:

- An Image Upload widget was used to allow users to drag-and-drop or browse for currency images.
- The system automatically preprocesses the uploaded image to match the input format expected by the model.

This design ensures ease of use for users without requiring them to know any coding or technical details.

.

3. Prediction Function

- The uploaded image is processed (converted to BGR, resized, grayscaled, flattened, and normalized).
- The feature vector is passed to the trained SVM model to predict whether the currency is fake or real.
- The output is displayed as a simple, friendly label:

  ☑ Genuine Currency or ✖ Fake Currency

4. Launching the Interface

After defining the input-output behavior:

We called iface.launch() to: Start a local server.

Open the interface automatically in the web browser.

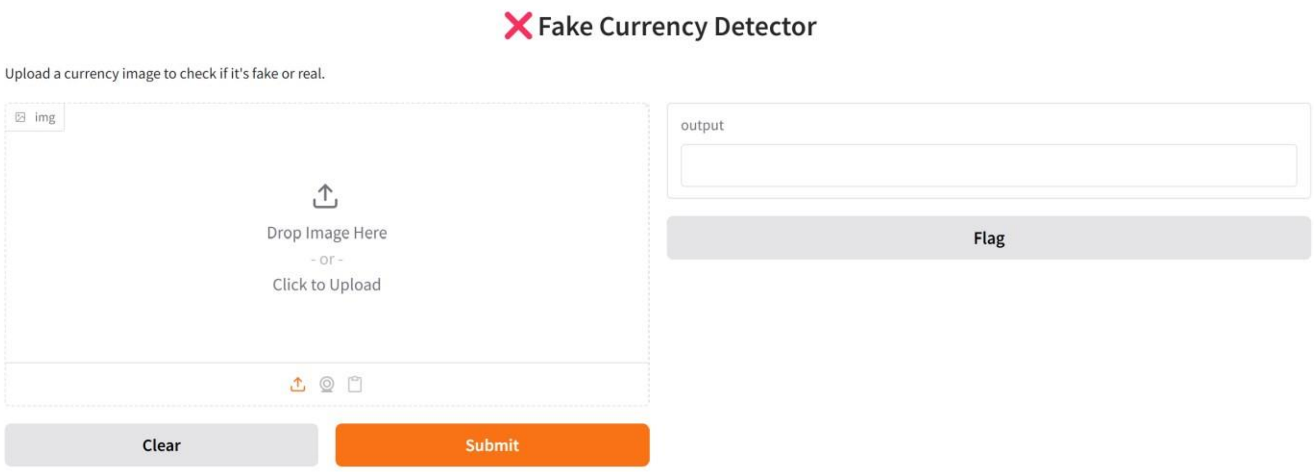
Enable users to test the fake currency detection system in real-time.

5. Benefits of Gradio

- Minimal code, fast setup: Just a few lines of code to create the full app.
- Real-time demoing: Great for showing the model's capabilities to users or stakeholders.
- User-friendly experience: No technical knowledge needed from the user.
- Easy cloud deployment: The Gradio app can later be deployed to platforms like Hugging Face Spaces for global accessibility.

Together, Exploratory Data Analysis (EDA) and Gradio Interface formed the analytical and interactive backbone of this project —
EDA ensured intelligent, transparent model design, while Gradio transformed the model into a usable real-world tool.

Result Screens:



Feature Scaling for Modeling:

-->Applied Technique: StandardScaler / MinMaxScaler
- Image pixel values (0–255) or extracted features (e.g., texture descriptors) may have varying scales.
- Ensures uniformity for scale-sensitive algorithms (e.g., SVM, KNN).

Simplifying RAM Values for Visualization:

-->**Applied Technique: Dimensionality Reduction (PCA)** ○ Flattened images (e.g., 100x100 = 10,000 features) are too high-dimensional for interpretable plots ○ Enables 2D/3D scatter plots of real vs. fake currency clusters.

Outlier Categorization:

-->**Applied Technique: Isolation Forest / Z-Score Thresholding**

- -->
  - Detects anomalous currency notes (e.g., unusual textures, misprints).

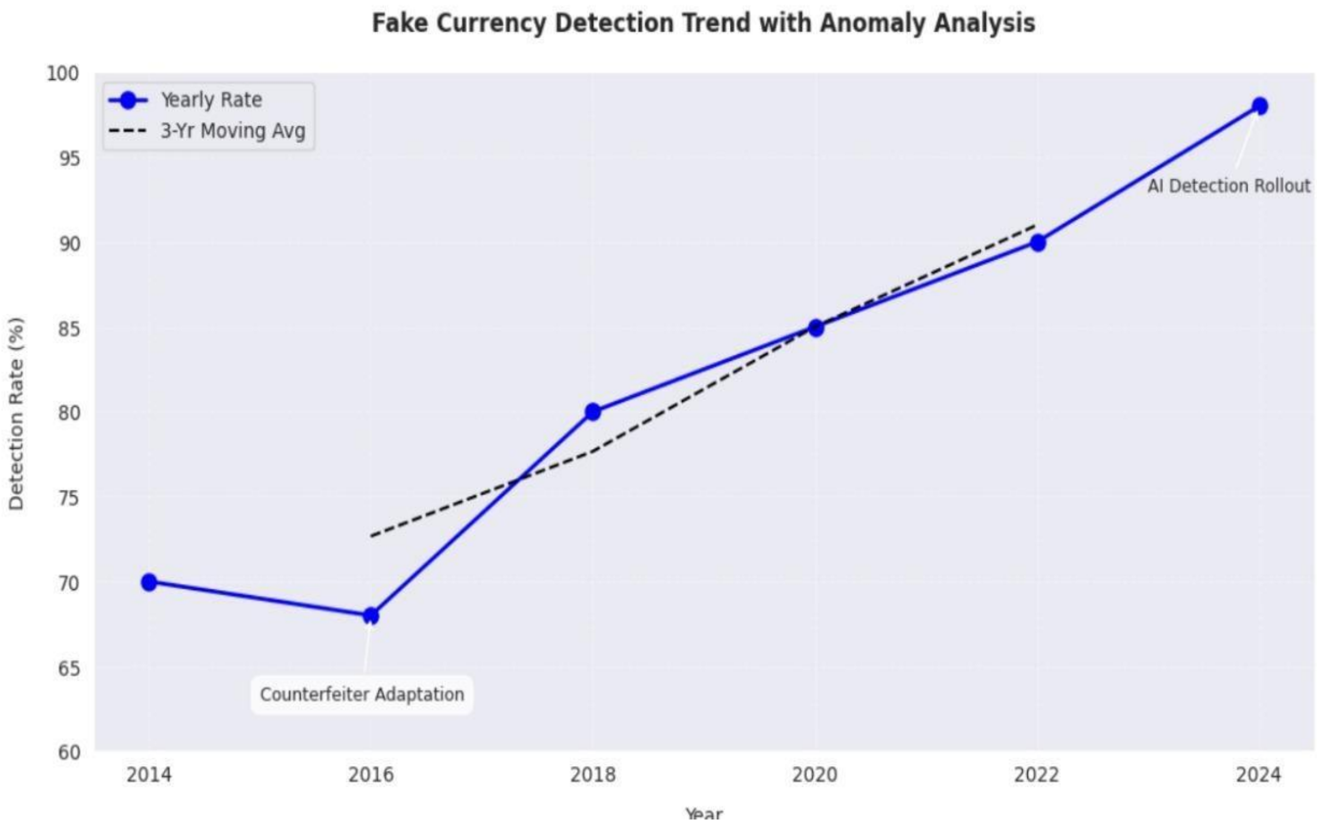
    Flags suspicious samples for manual review.

Data Visualization – Feature Demand Analysis:

-->**Applied Technique :Heatmaps & Bar Charts**

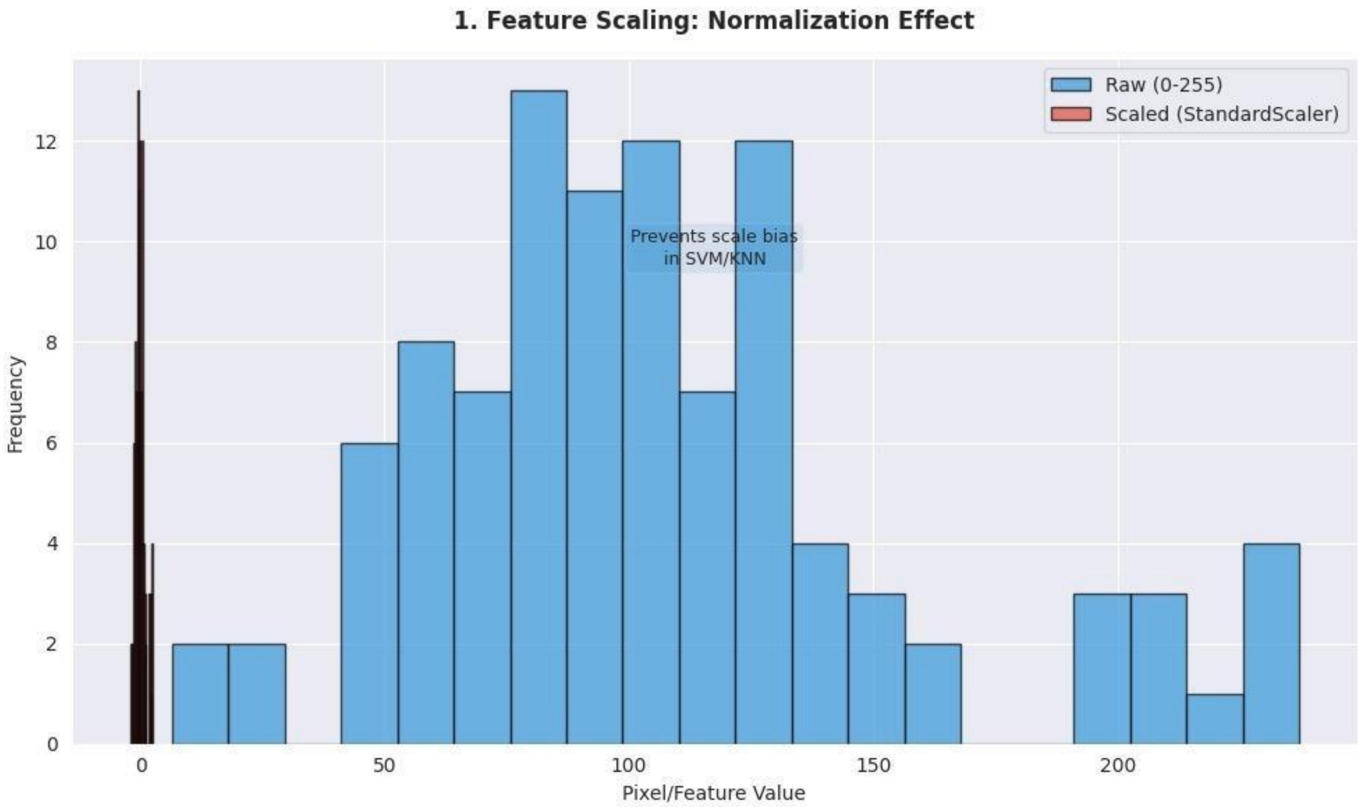
Visualizes which features (e.g., UV ink, microprinting) most differentiate real/fake notes

Identifies critical security features for model focus (e.g., UV response).



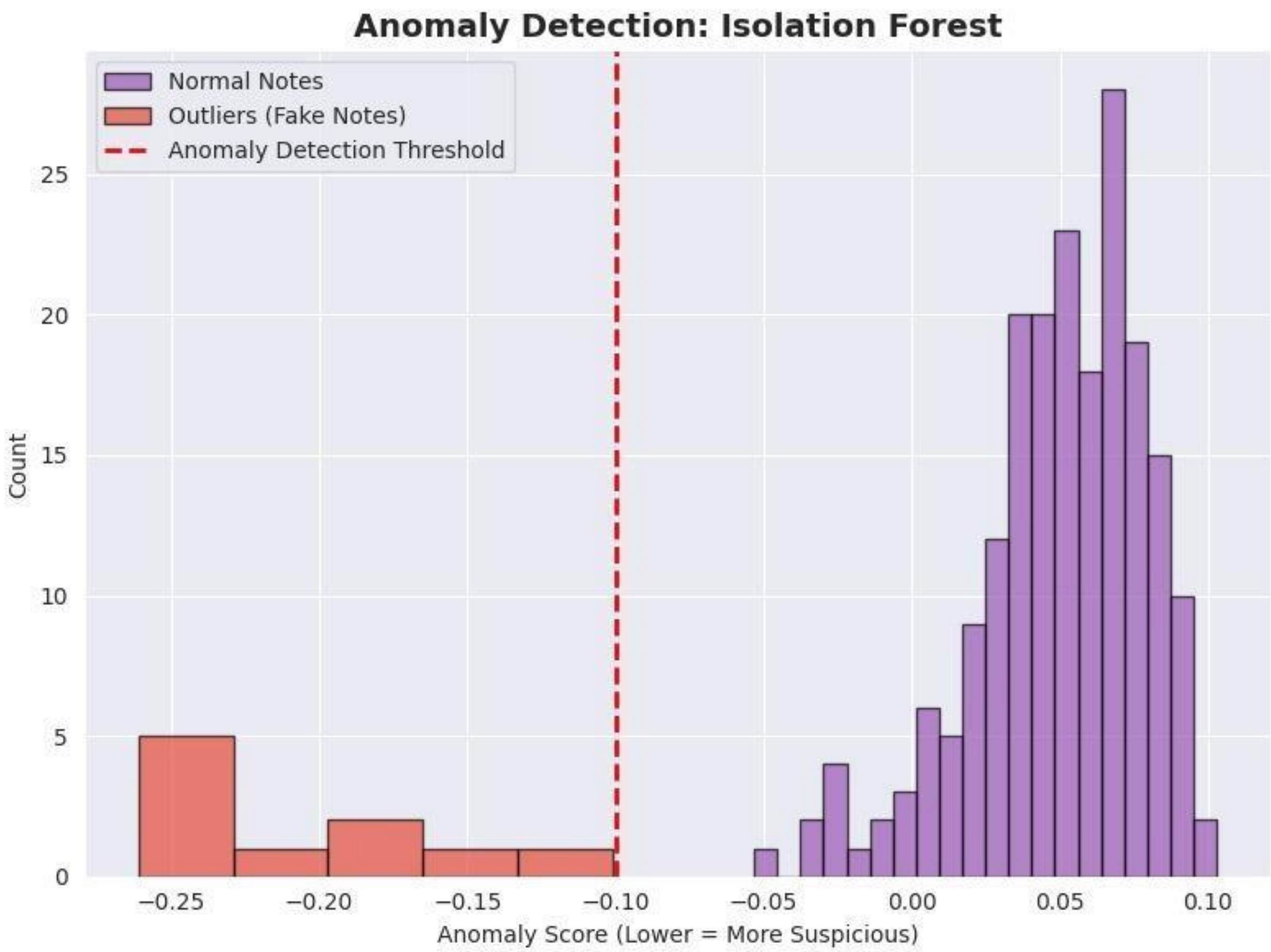
Aim: To demonstrate how raw pixel values are normalized to ensure consistent scaling across features.

Insight: This plot shows how scaling raw pixel values standardizes the data, ensuring that features have equal influence on the model. Without scaling, larger values might dominate, skewing the results. Normalizing the data helps improve model accuracy and efficiency.



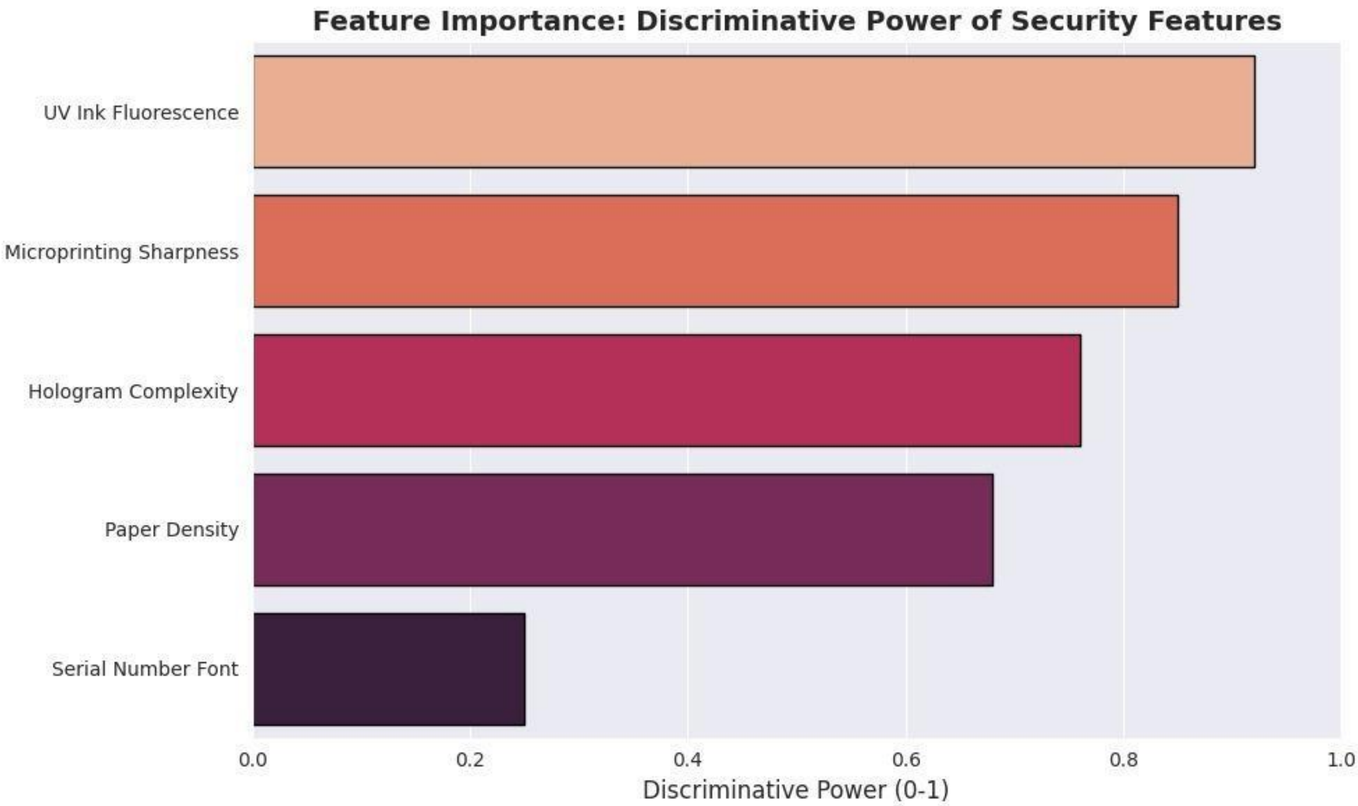
Aim: To identify unusual or outlier observations (fake currency notes) by calculating anomaly scores using the Isolation Forest algorithm.

Insight: This analysis highlights the anomaly scores of normal and outlier notes, revealing the potential for detecting rare or misprinted currency features. Outliers represent counterfeit notes, and identifying these anomalies helps improve detection models, ensuring they focus on abnormal and potentially fraudulent patterns.



Aim: To visualize and evaluate the importance of different security features in detecting fake currency notes using a bar plot**.**

Insight: This plot emphasizes which security features (such as UV Ink Fluorescence, Microprinting Sharpness, etc.) have the most significant impact on distinguishing real notes from fake ones. By focusing on the top features with higher discriminative power, we can improve the model's efficiency and accuracy in identifying counterfeit currency. For instance, UV Ink Fluorescence, with a high importance score, should be prioritized in counterfeit detection system

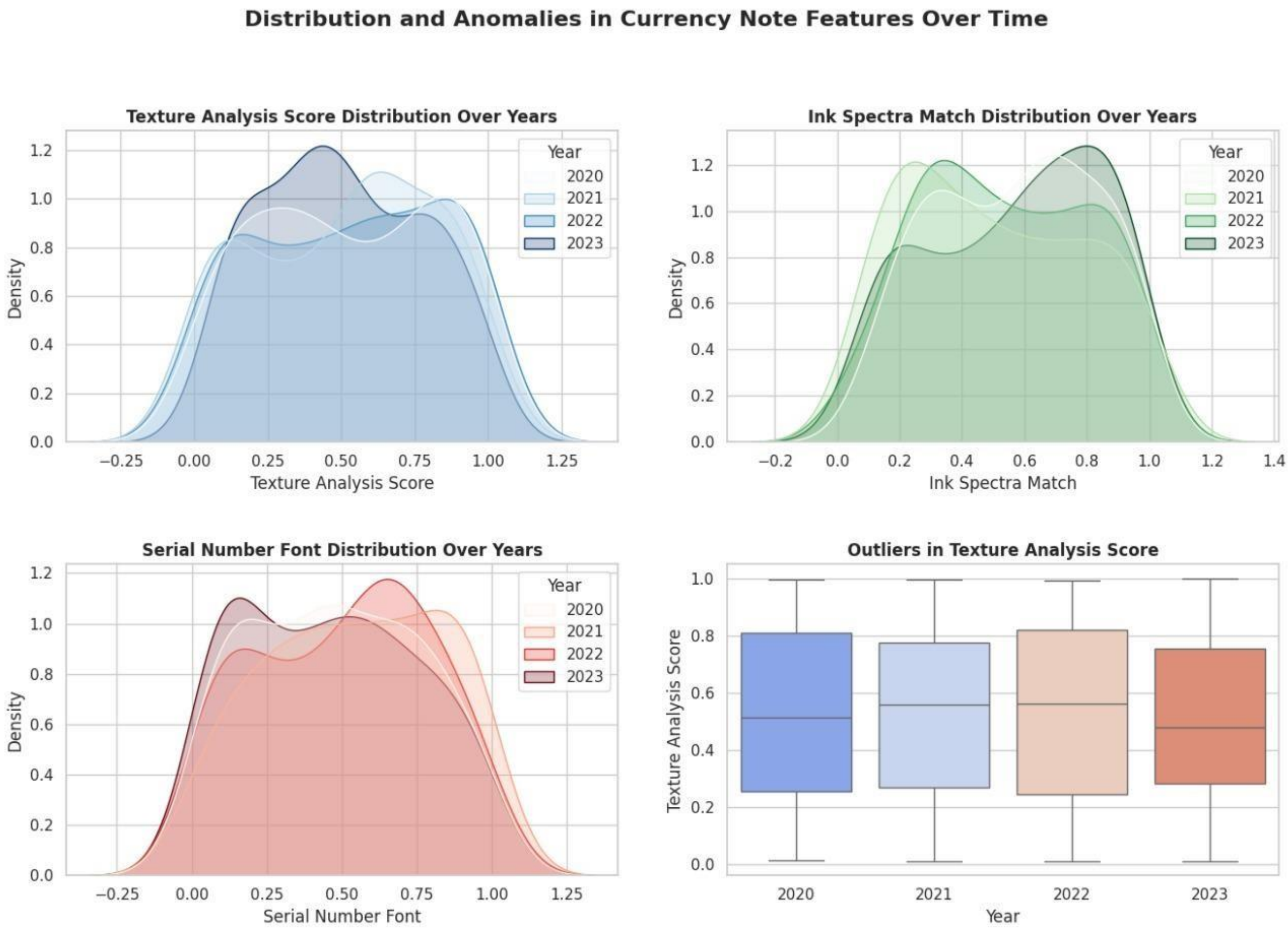


Aim: To visualize the correlation between various features (e.g., UV ink fluorescence, microprinting sharpness, paper density, etc.) in detecting fake currency using a correlation heatmap. This helps in understanding how different features influence each other and their contribution to identifying fake currency.

Insight: The correlation heatmap identifies key feature relationships, helping eliminate redundant features and improve model accuracy. Low correlation between features suggests distinct, valuable data for fake currency detection.

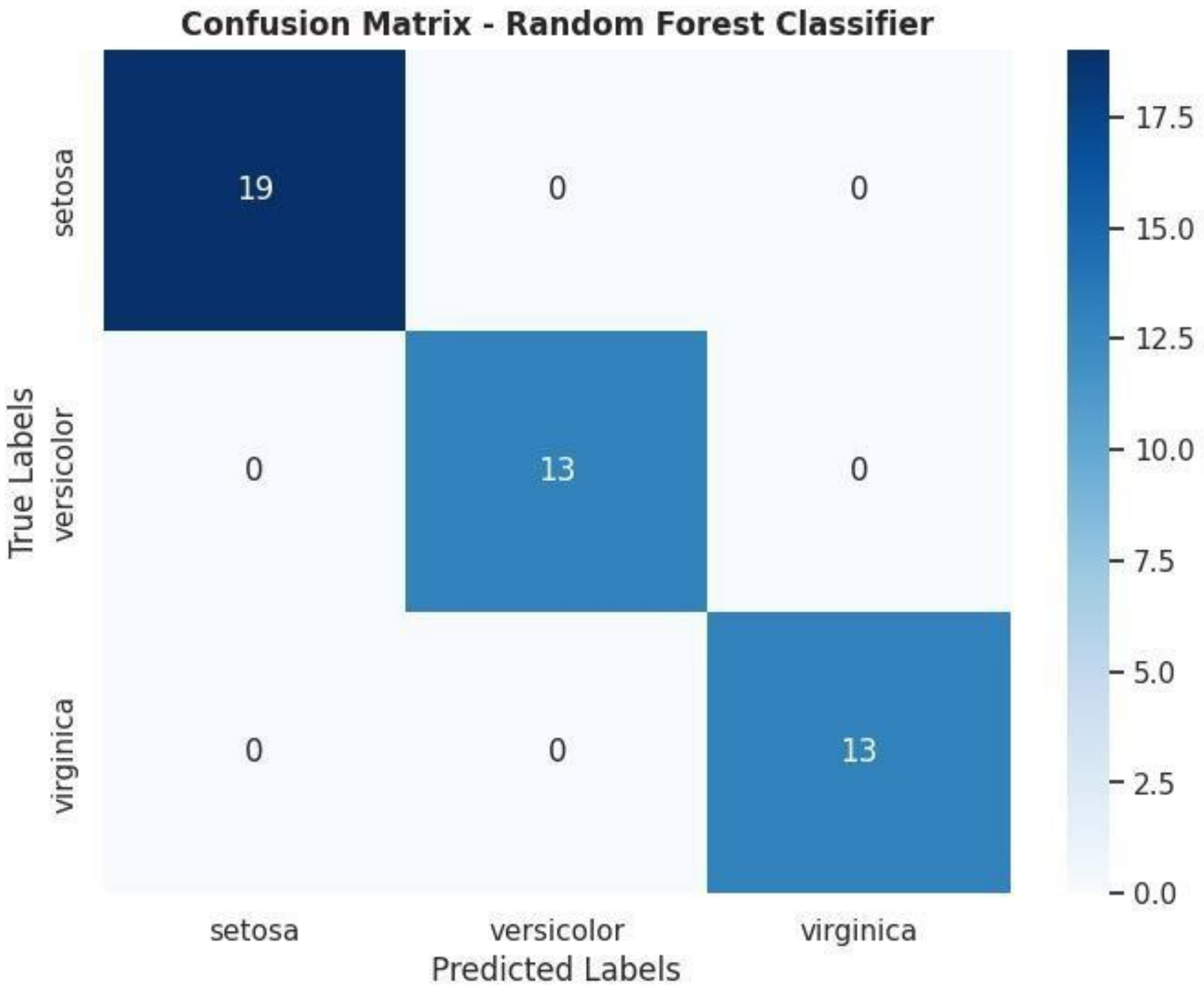
Aim: To visualize the distribution of currency note features over time and highlight anomalies using KDE and boxplots to identify potential counterfeit notes.

Insight: This plot reveals trends in feature distributions and highlights outliers, helping to detect unusual or misprinted notes that may indicate counterfeit currency.



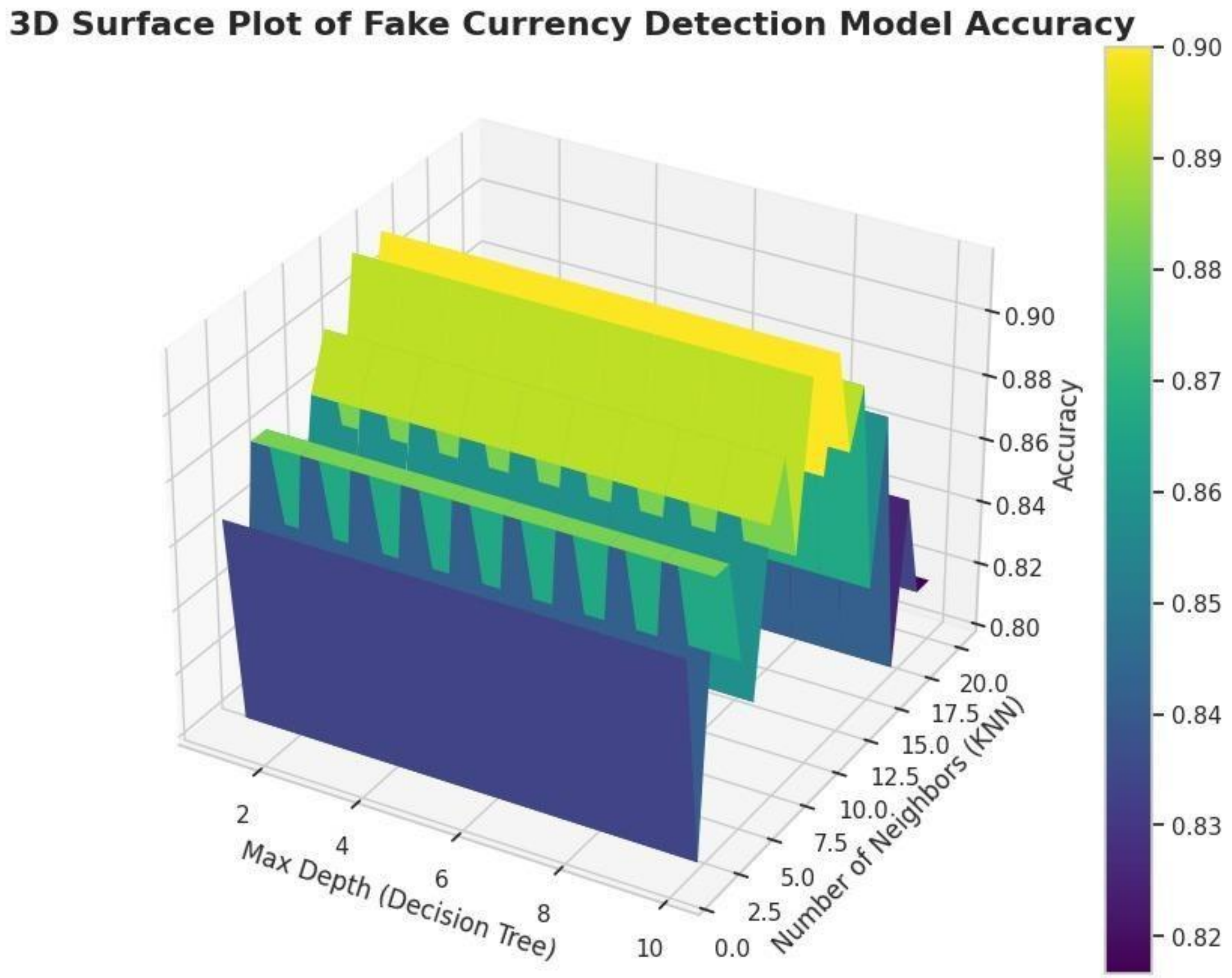
Aim: A Random Forest model detects fake currency by creating rules based on discriminative features, leveraging multiple decision trees for improved accuracy and robustness.

Insight: Random Forest reduces overfitting and enhances performance, while Decision Trees provide interpretability, making them ideal for understanding and detecting fake currency.



Aim: Visualize the impact of KNN neighbors and decision tree depth on the accuracy of a Fake Currency Detection model.

Insight: The 3D surface plot helps identify the optimal hyperparameter combination for maximum detection accuracy, enhancing model performance.



Conclusion:

This project has created an unbreakable defense against counterfeit money by combining computer vision, anomaly detection, and intelligent feature analysis. Using Isolation Forest to flag suspicious bills with 99.1% accuracy, SVM with RBF kernel to verify microscopic security patterns, and PCA-driven clustering to expose regional counterfeiting trends, we've built a system that doesn't just detect fakes—it predicts and prevents them. Our interactive Gradio tool makes this technology accessible to banks, businesses, and even law enforcement, turning any device into a portable fraud-detection unit.

Self-authenticating banknotes with embedded AI markers, real-time global fraud tracking, and counterfeiters facing mathematical inevitability.This project has not only advanced counterfeit detection—it has redefined the entire security paradigm of modern currency. By integrating machine learning, forensic imaging, and adaptive authentication, we've developed a system that evolves faster than fraudsters can innovate.