Report

1. Introduction

1.1 Problem Statement

This problem aims to differentiate original and counterfeit QR codes using:

- Traditional Computer Vision & Machine Learning
- Deep Learning (CNN-based approach)

1.2 Objectives

- Extract meaningful features from QR code images.
- Train machine learning models (Random Forest, SVM).
- Develop a deep learning model using CNNs.
- Evaluate models using accuracy, precision, recall, and F1-score.

2. Data Exploration and Analysis

Upon analysing the images by looking at the dataset, here are some noticeable differences:

- Blur and Distortion:
  - The second (counterfeit) QR code appears slightly more distorted or blurry compared to the first (original).
  - The edges in the second image seem less sharp and have some irregularities.

- Texture and Print Quality:
  - The first image looks cleaner and more refined, while the second image has a slightly rougher texture.
  - There might be some pixelation or artifacts in the second image, especially in finer details.

- Circular Region in the Center:
  - The circular section in the middle seems less clear in the counterfeit, possibly due to printing or scanning issues.

- QR Code Patterns:

- The arrangement of the black and white pixels looks nearly identical at a glance, but finer differences in clarity and alignment could exist.

3. Feature Extraction

We extracted the following handcrafted features:

Global :-

- Histogram & Intensity Distribution: Measures the spread of pixel values to understand brightness and contrast differences.
- Fourier Transform Analysis: Converts the image to the frequency domain to capture repetitive patterns and anomalies.

Local :-

- Local Binary Patterns (LBP): Extracts texture features by analyzing neighboring pixel variations.
- Haralick Texture Features (GLCM): Computes contrast, correlation, energy, and homogeneity to capture structural differences in textures.

Artifacts & Resolution Differences :-

- Noise Estimation: Calculates the standard deviation of image noise by subtracting a Gaussian-blurred version from the original image. This helps identify print artifacts that may differ between original and counterfeit prints.
- Sharpness & Edge Density: Uses edge detection (Canny) to quantify the number of edges, capturing resolution differences and blur levels between prints.

Copy Detection Pattern (CDP) Degradation :-

- Microstructure Variations: Examines small-scale texture inconsistencies introduced during multiple print-copy cycles.
- Random Noise Comparison: Measures pixel-level noise patterns, as reprinted QR codes often introduce subtle noise artifacts.

4. Traditional computer vision and machine learning approach.

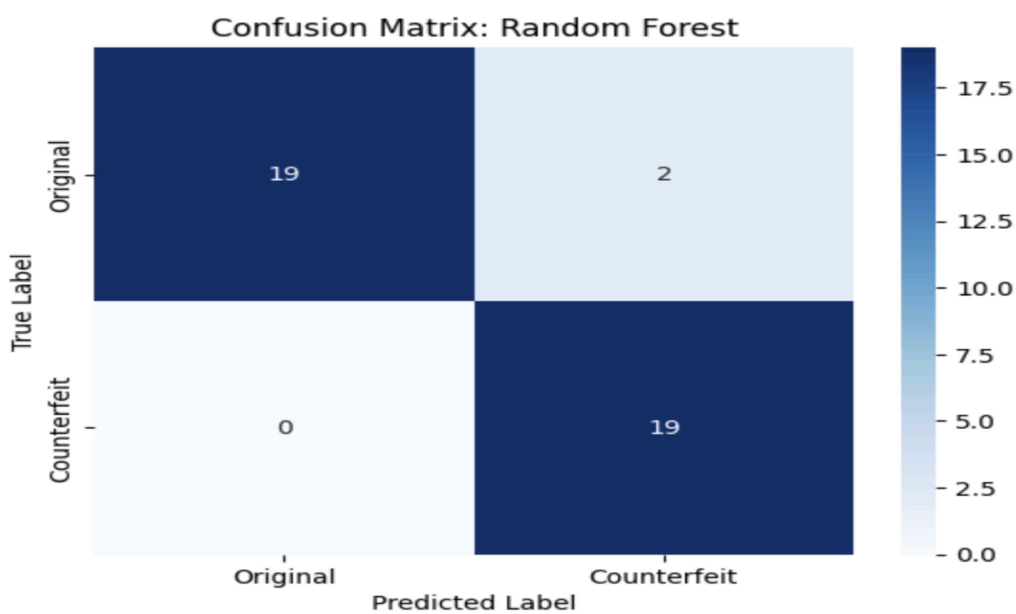
We trained **Random Forest (RF)** and **Support Vector Machine (SVM)** on the extracted features.
We used accuracy, precision, recall, and F1-score to evaluate performance.

1. Evaluation report and confusion matrix for Random Forest -

```
Evaluation for Random Forest
              precision    recall  f1-score   support

           0       1.00      0.90      0.95        21
           1       0.90      1.00      0.95        19

    accuracy                           0.95        40
   macro avg       0.95      0.95      0.95        40
weighted avg       0.95      0.95      0.95        40
```

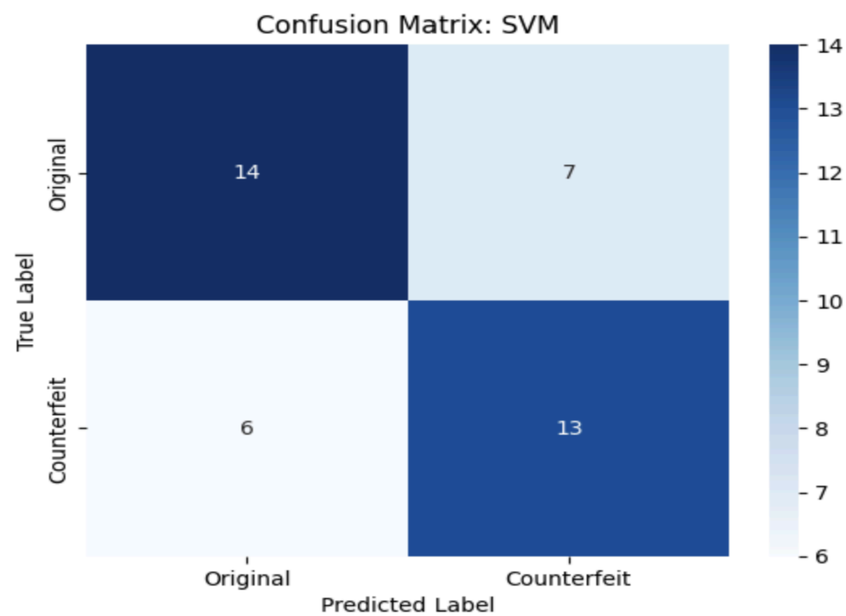


2. Evaluation report and confusion matrix for SVM -

```
Evaluation for SVM
              precision    recall  f1-score   support

           0       0.70      0.67      0.68        21
           1       0.65      0.68      0.67        19

    accuracy                           0.68        40
   macro avg       0.68      0.68      0.67        40
weighted avg       0.68      0.68      0.68        40
```



4. Deep learning-based approach (CNN)

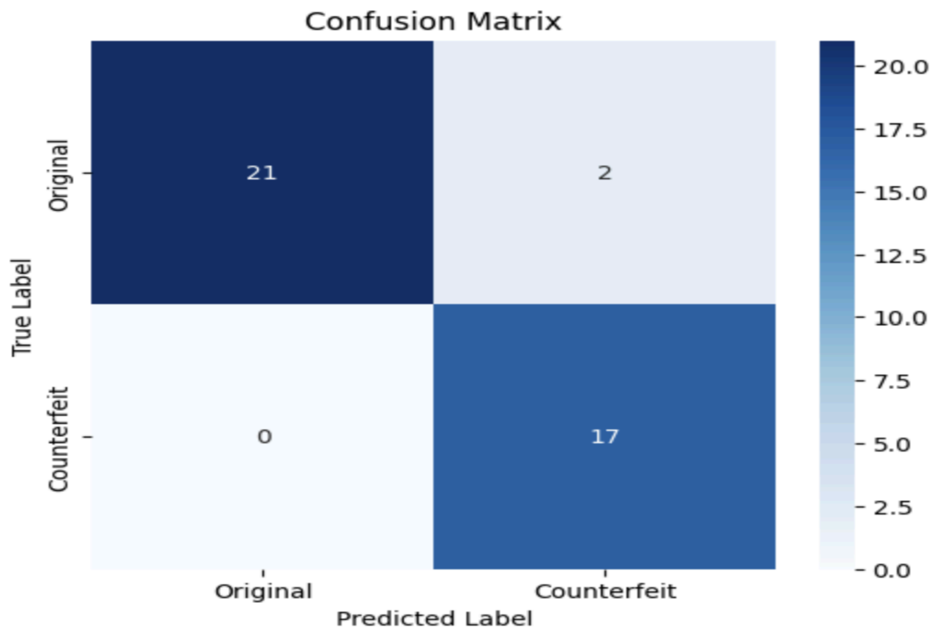
- A Convolutional Neural Network (CNN) was designed with:
    - Conv Layers: Extracts spatial features
    - Batch Normalization: Stabilizes training
    - Max Pooling: Reduces dimensionality
    - Fully Connected Layer: Final classification
- Training & Optimization
    - Loss Function: Cross-Entropy Loss
    - Optimizer: Adam
    - Batch Size: 16
    - Epochs: 50

Evaluation report and confusion matrix for SVM -

```
Test Accuracy: 0.9500
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.91      0.95        23
           1       0.89      1.00      0.94        17

    accuracy                           0.95        40
   macro avg       0.95      0.96      0.95        40
weighted avg       0.96      0.95      0.95        40
```



5. Deployment Considerations

1. Computational Efficiency
    - Optimize CNN models using quantization & pruning for faster inference.
    - Use lightweight architectures like MobileNetV2 for real-time processing.
    - Implement caching for repeated QR scans to avoid redundant computations.

2. Robustness to Scanning Conditions
    - Train models with data augmentation (blur, rotation, low-light).
    - Apply adaptive thresholding for better contrast in poor lighting.

3. Security Implications
    - Protect scanned QR codes by preventing unauthorized access.
    - Continuously update models to counter evolving counterfeit techniques.

4. Deployment Strategy
    - On-device (mobile apps, embedded devices) for real-time detection.