QR Code Authentication: Detecting Original vs. Counterfeit Prints

1. Introduction

QR codes are widely used for authentication, ticketing, and secure transactions. However, counterfeit QR codes pose a security risk. This project aims to classify original QR codes (first prints) and counterfeit QR codes (second prints) using both traditional machine learning and deep learning approaches.

2. Methodology

2.1 Data Exploration & Preprocessing

Dataset: Two folders containing original (first print) and counterfeit (second print) QR codes.

Preprocessing:

- Convert images to grayscale.
- Resize to 128x128 pixels.
- Normalize pixel values to [0,1].

2.2 Feature Engineering

- Sobel Edge Detection: Extracts gradient changes for detecting print quality differences.
- Histogram of Oriented Gradients (HOG): Captures local shape and texture differences.
- Flattened Feature Vector: Combines edge-based and intensity-based features.

2.3 Model Development

Traditional ML Model:

- Random Forest Classifier trained on extracted features.
- Feature Scaling: Standardized before training.

Deep Learning Model:

- CNN Architecture:
  - 3 Convolutional Layers for feature extraction.
  - MaxPooling Layers for dimensionality reduction.
  - Dense Layers for classification.

- Binary Crossentropy Loss with Adam Optimizer.

3. Experiments & Results

3.1 Performance Metrics

Model Performance:

Random Forest:

- Accuracy: 100%
- Precision: 1.00
- Recall: 1.00
- F1-score: 1.00

CNN:

- Accuracy: 100%
- Precision: 1.00
- Recall: 1.00
- F1-score: 1.00

3.2 Confusion Matrix

Random Forest:

[[21 0]
 [0 19]]

CNN:

[[21 0]
 [0 19]]

3.3 Misclassification Analysis

- Misclassified images are visualized for further analysis.
- Common errors occur due to low print quality or blurring.

4. Deployment Considerations

- Scanning Variations: Model should handle different lighting and scan resolutions.

- Computational Efficiency: CNN-based models may need optimization for real-time inference.
- Future Work:
  - Augment dataset with more variations.
  - Deploy model using Flask/FastAPI.

5. Conclusion

This study demonstrates an effective way to classify counterfeit QR codes. While traditional ML models perform well with feature engineering, CNNs provide higher accuracy by automatically learning distinguishing features. Future work involves dataset expansion and real-time deployment.