# QR Code Authentication System

- Shreya Paul

# QR Code Authentication System

## 1. Introduction

Counterfeit QR codes pose a major risk in various industries, including financial transactions, product authentication, and event ticketing. This project aims to develop a robust QR code authentication system that distinguishes between original and counterfeit QR codes using Machine Learning (ML) and Deep Learning (DL) approaches.

## 2. Data Exploration & Analysis

The dataset consists of QR code images categorized into:

- **Original (First Print) :** Authentic QR codes generated for legitimate use.
- **Counterfeit (Second Print) :** Fake QR codes with slight distortions.

By analyzing pixel intensity, texture, and structural variations, we aim to identify the key differences between original and counterfeit QR codes.

## 3. Feature Engineering

Feature extraction plays a critical role in distinguishing QR codes. We implemented the following techniques:

- **Local Binary Patterns (LBP) :** Captures texture variations and patterns.
- **Edge Detection (Canny Filter) :** Identifies structural changes in QR codes.
- **Statistical Features :** Measures pixel intensity variations.

These features are then used as input for Machine Learning models like SVM and Random Forest.

## 4. Model Development

We implemented two approaches:

- **Traditional Machine Learning (ML)**
- **Support Vector Machine (SVM) :** Classifies based on extracted features.
- **Random Forest :** An ensemble-based classifier for better generalization.
- **Deep Learning (CNN-based Approach):**
  - Convolutional Neural Network (CNN) is used to automatically extract features from QR codes.
  - The CNN model consists of multiple convolutional layers, max-pooling layers, and fully connected layers.
  - The CNN outperformed traditional ML models with an accuracy of 92%.

## 5. Evaluation & Results

We evaluated the models using the following metrics:

- **Accuracy:** Measures overall correctness of classification.

- **Precision & Recall:** Determines how well the model identifies counterfeit QR codes.

- **F1-score:** Balances precision and recall for a fair evaluation.

**Model Performance:**

- **SVM Accuracy:** 85%

- **Random Forest Accuracy:** 88%

- **CNN Accuracy:** 92%

Confusion matrices and other detailed evaluations are included in our analysis.

## 6. Deployment Considerations

For real-world implementation, we consider:

- **Model Deployment:** Convert the trained CNN model to ONNX or TensorFlow Lite for mobile and web applications.

- **Integration with APIs:** A Flask/Django-based API can be developed for authentication services.

- **Security Measures:** Prevent adversarial attacks by ensuring robust training with varied QR code samples.

## 7. Conclusion & Future Work

This project successfully classifies QR codes as either original or counterfeit, achieving high

accuracy using CNN models. Future improvements include:

- Expanding the dataset for better generalization.

- Enhancing model robustness against image distortions.

- Exploring transformer-based models for improved classification.