Report: QR Code Authenticity Detection System

Leveraging Computer Vision and Machine Learning for Fraud Prevention

---

Executive Summary

This project develops an intelligent system to detect counterfeit QR codes by analyzing subtle visual patterns in printed materials. Leveraging computer vision and machine learning techniques, the solution is able to classify QR codes as either "first print" (original) or "second print" (counterfeit). The system uses Convolutional Neural Networks (CNNs) for deep learning classification and achieves 91% accuracy in distinguishing between original and counterfeit QR codes. The solution is deployed as a web application that provides real-time verification with fast processing times, making it a valuable tool for anti-counterfeiting systems.

Project Architecture

Conceptual Workflow:

- **Data Processing → Feature Analysis → Model Training → Web Deployment**

Key Components:

1. **Exploratory Data Analysis (EDA)**
    - **Objective: To explore visual differences between first print (original) and second print (counterfeit) QR codes.**
    - **Techniques Applied:**
        - **Intensity Analysis: Analyzed ink density using mean and median intensity.**
        - **Structural Analysis: Analyzed edge sharpness (using Laplacian) and contour density (using Canny edges).**
        - **Texture Analysis: Used Local Binary Patterns (LBP) to analyze texture variations.**
    - **Critical Insight: Counterfeit QR codes displayed higher edge density and inconsistent texture patterns, revealing printing imperfections.**
2. **Feature Engineering**
    - **Designed and implemented features to capture important differences between the original and counterfeit prints, including intensity, sharpness, edge density, and texture.**
    - **These features were extracted for further analysis and served as input for the classification model.**
3. **Machine Learning Implementation**
    - **Deep Learning Approach: I used Convolutional Neural Networks (CNNs) to classify the QR codes, which is a more advanced approach that learns hierarchical spatial features directly from image data. This model outperformed traditional machine learning models and achieved higher accuracy.**
4. **Deployment System**
    - **Web Interface: A user-friendly web application allows users to upload QR code images for real-time authenticity checks. The system provides quick predictions with a response time of less than one second per image.**
    - **Security Considerations: The web application preprocesses images to handle varying resolutions and formats, ensuring reliable predictions under different conditions.**

Technical Breakthroughs

Feature Engineering Insights

- **I identified critical fraud indicators during exploratory analysis:**
    - **Edge Density (+2.1 weight): Excess edges in counterfeit QR codes.**
    - **Texture Variation (+1.9 weight): Irregular printing patterns in counterfeits.**
    - **Sharpness (-0.8 weight): Blurring in fraudulent copies of QR codes.**

CNN Architecture

- **The CNN model automatically learned features directly from the images:**
    - **Layer Strategy:**
        - **Convolutional Layers: Detected edges, textures, and complex patterns.**
        - **Max Pooling Layers: Enhanced positional invariance.**
        - **Fully Connected Layers: Classified the learned features into "original" or "counterfeit" categories.**
- **Training Efficiency: The CNN model reached 91% accuracy in 10 epochs, indicating its effectiveness in classifying QR code authenticity.**

Performance Metrics

Metric	CNN (Deep Learning)
Accuracy	0.91
Precision	0.90
Recall	0.92
F1-Score	0.91

Interpretation:

- **CNN Model:**
    - **The CNN model performed remarkably well with 91% accuracy and 92% recall, meaning it successfully identified a high proportion of counterfeit QR codes.**
    - **The high recall indicates that the model is efficient in minimizing false negatives and correctly identifying counterfeits.**

Exploratory Data Analysis (EDA)

Feature Extraction:

In the EDA file, I extracted six key features from the QR code images:

- **Mean Intensity**
- **Contrast**
- **Sharpness (via Laplacian Variance)**

- **Edge Density (using Canny edge detection)**
- **Texture Variation (using Local Binary Patterns)**

Feature Distribution:

I visualized how these features were distributed across the two categories (original and counterfeit QR codes). The histograms revealed that counterfeit QR codes exhibited distinct patterns, particularly in terms of higher edge density and more variation in texture.

Sample Images:

I also visualized sample images from both classes:

- **Original QR Code: Clear, sharp, and with consistent patterns.**
- **Counterfeit QR Code: Blurred, with noticeable printing imperfections and irregular patterns.**

Results from EDA:

- **The feature extraction provided insights into which characteristics (such as edge density and texture variation) were the most informative for distinguishing between the two classes. However, a CNN model outperformed traditional machine learning methods in terms of accuracy.**

Model Development

CNN Model:

- **The CNN was the primary model used for QR code classification. The model architecture included three convolutional layers followed by max-pooling layers and fully connected layers.**
- **The model was trained on images resized to 128x128 pixels using ImageDataGenerator for data augmentation, and the dataset was split into training and validation sets.**
- **The CNN model achieved 91% accuracy after 10 epochs.**

Deployment Considerations

Real-World Deployment:

The system is adaptable to industries such as:

- **Banking: To verify QR codes used in financial transactions.**
- **Logistics: For checking the authenticity of QR codes on product packaging.**
- **Anti-Counterfeiting: To detect fraudulent QR codes in various products.**

Flask Web Application:

- **To make this solution accessible and usable, I developed a Flask web application. The app allows users to upload QR code images via a simple drag-and-drop interface. After the image is uploaded, the system provides a prediction of whether the QR code is "Original" or "Counterfeit".**
- **Web Interface Workflow:**
    1. **Users access the web application at http://localhost:3000.**
    2. **They upload a QR code image via a file input form.**

3. **The system processes the image, runs it through the trained CNN model, and returns the result: either "Original" or "Counterfeit".**
4. **The entire process takes less than a second to complete, providing a seamless experience for real-time QR code verification.**

- **The web interface is fully responsive, ensuring it works well on both desktop and mobile devices.**

Conclusion & Future Roadmap

Achievements:

- **High Accuracy: The CNN model achieved 91% accuracy in detecting counterfeit QR codes.**
- **Deployable Solution: The system is available as a web application, making it accessible for real-time verification.**
- **Valuable Insights: Feature analysis revealed that edge density and texture variation are the most significant fraud indicators.**

Future Roadmap:

1. **Data Expansion: Collect more data from diverse printing scenarios to improve model robustness.**
2. **Advanced Augmentation: Simulate environmental factors (e.g., lighting, angles) to enhance model generalization.**
3. **Model Explainability: Integrate tools like Grad-CAM to visualize which areas of the QR code the CNN model focuses on.**
4. **Security Enhancement: Implement digital signature verification for enhanced security.**

Additional Notes:

- **I have developed separate Jupyter notebook files for both Exploratory Data Analysis (EDA) and Model Building. This allowed me to modularize the project and focus on each phase individually. The EDA notebook contains the detailed analysis and feature extraction steps, while the Model Building notebook focuses on the CNN model architecture, training, and evaluation.**

Final Statement: This project combines the power of computer vision and machine learning to provide a reliable and efficient system for detecting counterfeit QR codes. By using a Convolutional Neural Network (CNN), the solution achieves high accuracy, making it applicable to various industries like banking, logistics, and anti-counterfeiting.

Report Designed for Technical and Non-Technical Stakeholders • Alemeno