**QR Code Authentication: Original vs. Counterfeit Detection**

**Introduction**

Ensuring the authenticity of QR codes is crucial in preventing counterfeit products and fraudulent activities. In this project, we explore a machine learning-based approach to differentiate between original and counterfeit QR codes using feature extraction techniques and classification models. Our methodology involves image preprocessing, feature extraction, model training, and performance evaluation to achieve accurate classification.

**Methodology**

**Data Preprocessing**

1. **Loading Images**: The dataset consists of images labeled as "First Print" (original) and "Second Print" (counterfeit). The images were loaded, converted to grayscale, and resized to 256x256 pixels for uniformity.
2. **Normalization**: Pixel values were scaled to the range [0,1] to improve model convergence.
3. **Splitting the Dataset**: The dataset was divided into training (80%) and testing (20%) sets while ensuring class balance.

**Feature Extraction**

To improve classification performance, we extracted meaningful features from the images using:

1. **Histogram of Oriented Gradients (HOG)**: Captures edge and texture information.
2. **Local Binary Patterns (LBP)**: Extracts texture descriptors.
3. **Gabor Filters**: Enhances feature representation by capturing frequency and orientation.

The extracted features were standardized using **StandardScaler** to ensure consistency across different feature types.

**Model Training and Evaluation**

We experimented with three classification models:

**Support Vector Machine (SVM)**

* A linear SVM classifier was trained using extracted features.
* **Results**:
  + Accuracy: 100%
  + Confusion Matrix:

|  | **Predicted 0** | **Predicted 1** |
| --- | --- | --- |
| **Actual 0** | 20 | 0 |
| **Actual 1** | 0 | 20 |

**XGBoost Classifier**

* XGBoost was applied to enhance classification performance.
* **Results**:
  + Accuracy: 97%
  + Confusion Matrix:

|  | **Predicted 0** | **Predicted 1** |
| --- | --- | --- |
| **Actual 0** | 19 | 1 |
| **Actual 1** | 0 | 20 |

**Convolutional Neural Network (CNN)**

* A CNN model was designed with multiple convolutional layers followed by fully connected layers.
* **Architecture**:
  + Conv2D (32 filters, 3x3 kernel) + MaxPooling2D
  + Conv2D (64 filters, 3x3 kernel) + MaxPooling2D
  + Conv2D (128 filters, 3x3 kernel) + MaxPooling2D
  + Fully connected dense layers with Dropout
* **Results**:
  + Accuracy: 95%
  + Confusion Matrix:

|  | **Predicted 0** | **Predicted 1** |
| --- | --- | --- |
| **Actual 0** | 19 | 1 |
| **Actual 1** | 1 | 19 |

**Practical Considerations for Real-World Deployment**

For real-world applications, deploying a QR code authentication model requires:

1. Efficiency: Optimized models for real-time detection, ensuring low latency.
2. Scalability: Ability to handle large-scale verification in diverse environments.
3. Robustness: Ensuring the model generalizes well to different print qualities and environmental variations.
4. Integration: Seamless deployment through APIs or mobile applications for ease of access.
5. Security: Preventing adversarial attacks and tampering with QR codes.

**Conclusion**

This study demonstrated that machine learning and deep learning techniques can effectively differentiate between original and counterfeit QR codes. The **SVM model achieved perfect accuracy**, while **XGBoost and CNN performed slightly lower** but still exhibited strong performance. Future improvements could include:

* **Incorporating more advanced deep learning models** for better generalization.
* **Using data augmentation** to enhance model robustness.
* **Exploring additional feature extraction techniques** to capture intricate details.

By leveraging these techniques, QR code authentication can be significantly improved, ensuring greater security in real-world applications.