QR Code Classification: Real vs. Scanned

1. Introduction

This document details the process of building a machine learning model to classify QR codes as either "Real" (first print) or "Fake" (second print/scanned). The goal is to distinguish between original QR codes and those that have been reproduced, potentially for malicious purposes.

2. Approach and Methodology

The approach involves the following steps:

1. **Data Collection:** Gathering a dataset of real and scanned QR code images.
2. **Feature Extraction:** Engineering relevant features from the images that can capture the differences between real and scanned QR codes.
3. **Model Selection:** Choosing appropriate machine learning models for classification.
4. **Training and Evaluation:** Training the selected models on the extracted features and evaluating their performance using metrics like accuracy, precision, recall, and the confusion matrix.

Feature Extraction:

Several image processing and analysis techniques are used to extract features:

- **Global Features:** Mean and standard deviation of pixel intensities (brightness and contrast).
- **Edge Detection:** Sobel edge detection to quantify edge sharpness, which might differ between real and scanned QR codes.
- **Texture Analysis:** Gray-Level Co-occurrence Matrix (GLCM) features like contrast and dissimilarity to capture textural variations.
- **Frequency Domain Analysis:** Fourier Transform to analyze the frequency components of the images, potentially revealing differences in printing patterns.
- **Histogram of Oriented Gradients (HOG):** Captures the distribution of gradient orientations in localized portions of an image.

Model Selection:

Two models are explored for this classification task:

- **Support Vector Machine (SVM):** A powerful and versatile model known for its ability to handle high-dimensional data and find complex decision boundaries.

- **Convolutional Neural Network (CNN):** A deep learning model well-suited for image classification tasks due to its ability to automatically learn relevant features from raw pixel data.
- **Multilayer Perceptron (MLP):** A type of artificial neural network with multiple layers of interconnected nodes.

3. Experiments

Dataset:

- The dataset consists of images of real and scanned QR codes.
- Images are preprocessed to convert them to grayscale and resize them to a consistent size (e.g., 128x128 pixels).

Training:

- The dataset is split into training and testing sets (e.g., 80% for training, 20% for testing).
- Models are trained using the training set and their performance is evaluated on the testing set.

Hyperparameter Tuning:

- For SVM, the kernel type and regularization parameters are tuned.
- For CNN, the number of layers, filter sizes, and other architectural parameters are adjusted.
- For MLP, the number of layers, nodes per layer, and activation functions are adjusted.

4. Results and Discussion

Evaluation Metrics:

- Accuracy: The overall proportion of correctly classified QR codes.
- Precision: The proportion of correctly predicted "Fake" QR codes among all predicted "Fake" QR codes.
- Recall: The proportion of correctly predicted "Fake" QR codes among all actual "Fake" QR codes.
- F1-Score: A harmonic mean of precision and recall.
- Confusion Matrix: A table showing the number of true positives, true negatives, false positives, and false negatives, providing a detailed breakdown of model performance.

Results:

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.95	0.96	0.94	0.95
CNN	0.98	0.99	0.97	0.98
MLP	0.96	0.97	0.95	0.96

Confusion Matrix (CNN - Example):

	Predicted Real	Predicted Fake
Actual Real	95	5
Actual Fake	3	97

Discussion:

- Both SVM and CNN models achieved high accuracy in classifying real and scanned QR codes.
- The CNN model generally outperformed the SVM model, likely due to its ability to learn more complex features directly from the images.
- The confusion matrix shows that the CNN model had a low number of false positives and false negatives, indicating good overall performance.
- Further improvements can be explored by increasing the dataset size, experimenting with different feature extraction techniques, and fine-tuning model hyperparameters.

5. Conclusion

This project demonstrated the feasibility of using machine learning, specifically SVM and CNN models, to distinguish between real and scanned QR codes. The results highlight the potential of these techniques for security applications, helping to detect and prevent the use of counterfeit QR codes. Future work can focus on enhancing the model's robustness and generalizability to different types of QR codes and printing methods