QR Code Authentication: Detecting Original vs. Counterfeit Prints

Background

In this take-home assignment, you will work on a classification task to identify original vs. counterfeit QR codes. The dataset contains QR codes with embedded copy detection patterns (CDPs) that help distinguish between original prints (first prints) and counterfeit copies (second prints). These second prints are created by scanning and reprinting the original QR codes.

Task Overview

Your goal is to develop a machine learning model that can accurately classify QR code images as either "first print" (original) or "second print" (counterfeit). This technology is crucial for anti-counterfeiting systems where verifying the authenticity of printed QR codes is essential.

Dataset Description

The dataset consists of:

- **First prints**: Original QR codes with embedded copy detection patterns
- **Second prints**: Counterfeit versions created by scanning and reprinting first prints

Each image in the dataset contains a QR code with subtle differences in print quality, microscopic patterns, and other features that distinguish originals from counterfeits.

Dataset Link - https://drive.google.com/drive/folders/1pPeWT1zntlKXnuY_yHmpI-ZzKl4nLgQS?usp=drive_link

Assignment Requirements

1. **Data Exploration and Analysis**
    - Analyze the provided dataset and identify key visual differences between first and second prints
    - Extract and visualize relevant features that could help in classification
    - Report statistics about the dataset composition.

2. **Feature Engineering**
    - Design and implement features that capture the differences between original and counterfeit prints
    - Consider both global image properties and local patterns within the QR codes
    - Explore techniques that focus on print artifacts, resolution differences, and CDP degradation
3. **Model Development**
    - Implement at least two different classification approaches:
        - A traditional computer vision and machine learning pipeline
        - A deep learning-based approach (CNN or similar architecture)
    - Train your models using appropriate validation strategies
    - Document your implementation choices and reasoning
4. **Evaluation and Results**
    - Evaluate your models using appropriate metrics (accuracy, precision, recall, F1-score, etc.)
    - Analyze misclassifications and provide insights into challenging cases
    - Compare the performance of your different approaches
5. **Deployment Considerations**
    - Discuss how your solution could be deployed in a real-world setting
    - Address considerations like computational efficiency, robustness to different scanning conditions, and security implications

Deliverables

1. **Code**: A well-organized repository with all code used for the assignment
2. **Report**: A comprehensive document (4-6 pages) explaining your approach, methodology, experiments, and results along with confusion_matrix , training metrics etc.

Evaluation Criteria

Your submission will be evaluated based on:

- Technical depth and correctness of your implementation
- Quality of your feature engineering and model selection process
- Performance of your classification models
- Clarity of your code, documentation, and explanations
- Creativity and thoughtfulness in addressing the problem
- Practical considerations for real-world deployment

Instructions for Getting Started

1. Download the dataset from the provided link
2. Set up your development environment with the necessary libraries
3. Explore the images to understand the classification challenge
4. Begin with simple analysis before moving to more complex models

Timeline

You have 5 days to complete this assignment from the time you receive it. Please submit all deliverables by <deadline>.