Report: Currency Note Template Matching using OpenCV

---

1. Introduction to Template Matching

Template matching is a computer vision technique used to locate and match a sub-image (called the template) within a larger image. It works by sliding the template across the input image and computing a similarity score at each position. The highest-scoring position is considered the best match.

---

2. Selected Template and Methodology

Selected Template

I selected Quaid-e-Azam's face from currency note as the template. This region is visually distinct and appears consistently on all notes, making it a reliable anchor point.

Methodology

- **Preprocessing:**
    - Converted images to grayscale.
    - Applied cv2.equalizeHist() to improve contrast.
    - Used Gaussian Blur to reduce minor noise while preserving structure.

- **Template Matching:**
    - Used cv2.matchTemplate() with cv2.TM_CCOEFF_NORMED for similarity scoring.
    - Implemented multi-scale and multi-angle matching by resizing and rotating the template.
    - Extracted the best match location and template size.
- **Bounding Boxes:**
    - A green rectangle was drawn around the matched face.
    - Two blue rectangles were drawn to cover the top-right and bottom-left identification numbers based on relative offsets from the face region.
    - Offsets and box sizes were scaled dynamically using the matched template size.
    - Bounding boxes were slightly shrunk using a custom adjustBox() function to better fit the actual content.
- **Output:**
    - Annotated images were saved.
    - Coordinates of the ID boxes were saved in a CSV file.

---

3. Challenges and Solutions

Challenge	Description	Solution
Varying image brightness	Some note images were overly bright, reducing matching accuracy	Applied cv2.equalizeHist() to improve contrast
Different scales	Notes were captured at various resolutions and zoom levels	Implemented template matching over multiple scales (0.9x, 1.0x, 1.1x)
Rotation inconsistency	Some notes were slightly rotated	Used rotation-invariant matching with angles from -90° to +90°
Inaccurate ID box positioning	Using fixed pixel offsets didn't work across different note sizes	Switched to relative offsets based on template width and height
Box overflow	Boxes sometimes extended out of the image bounds	Used adjustBox() to shrink and clamp boxes within image dimensions

4. Results

Output images are also attached.
