1. **Đôi nét về OpenCV**

Project [OpenCV](https://opencv.org/) được bắt đầu từ Intel năm 1999 bởi Gary Bradsky. OpenCV viết tắt của Open Source Computer Vision Library. OpenCV là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning, và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo real-time.

OpenCV là 1 thư viện mã nguồn mở miễn phí cho cả học tập và sử dụng với mục đích thương mại. Do đó nó có một cộng đồng người dùng khá hùng hậu hoạt động trên khắp thế giới bởi nhu cầu cần đến nó ngày càng tăng theo xu hướng chạy đua về sử dụng computer vision của các công ty công nghệ.

OpenCV được sử dụng cho đa dạng nhiều mục đích và ứng dụng khác nhau, và nhóm em đã sử dụng thư viện này xây dựng 1 công cụ chấm điểm bài thi trắc nghiệm

1. **Sau đây sẽ là phần nội dung chính**

Và bây giờ nhóm em sẽ trình bày các bước để có thể xử lí được ảnh bài thi trắc nghiệm

* Đầu tiên : Nhập dữ liệu đầu vào

Nhập ảnh bài thi (file ảnh scan)

Nhập file đáp án đúng

* Bước 2 : Tiền xử lí ảnh

Áp dụng các bộ lọc (Gaussian Blur, Median Blur).

Chuyển đổi ảnh sang dạng nhị phân (Binary Threshold).

Xử lý để làm nổi bật đường viền và cạnh (Edge Detection).

* Bước 3 : **Phát hiện điểm mốc (Anchor Point Detection):**

**Sử dụng các hàm**

get\_contours\_rect, get\_contours\_circle, get\_anchor

Tìm và sắp xếp các đường viền chính.

Xác định các điểm mốc để hiệu chỉnh ảnh: Điểm mốc vùng mã đề và vùng đáp án trắc nghiệm.

* Bước 4 : **Biến đổi phối cảnh**

Các hàm : four\_point\_transform, \_\_two\_point\_transform để tìm ra các điểm mốc, qua đó để

* + Hiệu chỉnh lại ảnh tổng quan.
  + Cắt vùng ảnh cần thiết: mã đề, Số báo danh, các câu trắc nghiệm.
* Bước 5 : **Nhận diện đáp án (Answer Detection):**

matrix2ans, find\_correct\_ans

Nhận diện vùng đáp án

Đếm pixel trắng trong mỗi vùng để xác định câu trả lời được chọn.

Chuyển đổi dữ liệu ma trận thành các lựa chọn ABCD.

* Bước 6 : **Chấm điểm (Grading and Scoring):**

Hàm grade để:

So sánh câu trả lời với đáp án đúng.

Tính điểm dựa trên tỷ lệ số câu trả lời đúng và tổng số câu.

* Bước 7 : **Xuất kết quả (Output Results):**

Hiển thị số điểm đạt được.

Tô màu câu trả lời đúng (xanh) và sai (đỏ) trên ảnh.

Xuất thông tin người thi và kết quả bài làm.

1. Sau đây sẽ là phần demo :

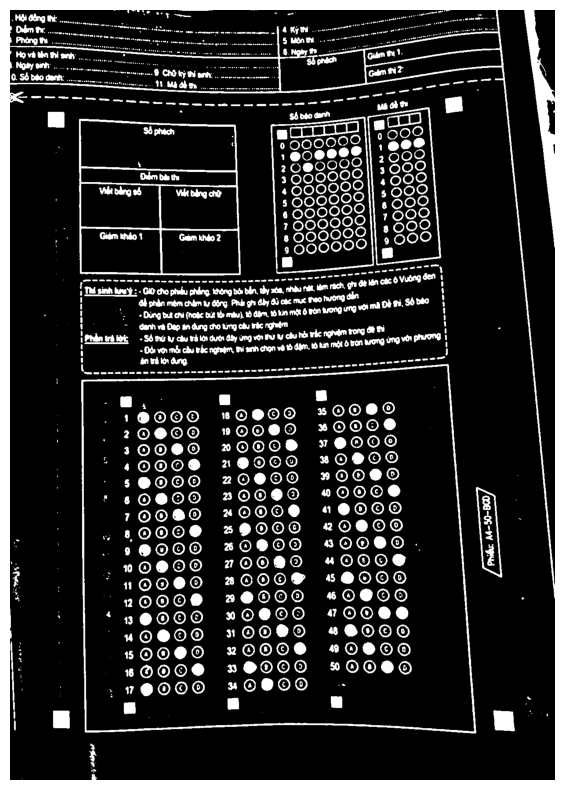
**END**

**1. Khởi tạo và chuẩn hóa ảnh**

* **Lớp Grader**
  + **Hàm \_\_init\_\_**
    - Nhận ảnh gốc (img), danh sách đáp án đúng (test\_answers), và điểm tối đa (max\_score).
    - Resize ảnh về kích thước cố định (1240x1754) tương ứng với khổ A4 (150 PPI).
    - Lưu ảnh gốc để xử lý và hiển thị kết quả.
  + **Ý nghĩa:** Chuẩn hóa kích thước ảnh đảm bảo tất cả các bài thi đều có kích thước tương tự, thuận tiện cho việc phát hiện và xử lý các vùng trong ảnh.

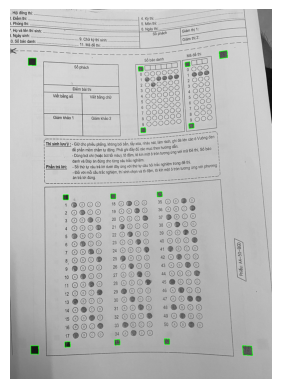
**2. Tiền xử lý ảnh**

Các bước này làm nổi bật thông tin cần thiết từ ảnh:

* **\_\_preprocess\_contour**
  + Làm mịn ảnh bằng bộ lọc trung bình và Gaussian để giảm nhiễu.
  + Áp dụng ngưỡng thích ứng (cv.adaptiveThreshold) để tạo ảnh nhị phân, làm nổi bật các vùng cần phát hiện như hình vuông.
  + 
* **\_\_preprocess\_detect**
  + Tương tự \_\_preprocess\_contour, nhưng chuyên biệt để nhận diện các hình tròn hoặc câu trả lời tô.
* **\_\_preprocess\_edge**
  + Áp dụng bộ lọc cạnh Canny để phát hiện biên của các đối tượng.

**Mục đích:** Chuẩn bị ảnh cho các bước phát hiện contour và đối tượng hình học.

**3. Xác định các điểm neo (Anchor Points)**

* **\_\_get\_contours\_rect**
  + Dò contour của các hình vuông trong ảnh.
  + Sử dụng các tiêu chí về tỷ lệ (aspect ratio) và số lượng điểm ảnh trắng (min\_white\_pixels) để lọc ra hình vuông hợp lệ.
  + Sắp xếp các hình vuông theo diện tích giảm dần.
* **\_\_get\_anchor**
  + Tính toán tọa độ trung bình của các hình vuông để tìm các điểm neo.
  + Kết quả: Các điểm neo cần thiết để thực hiện phép biến đổi.
  + 

**4. Phép biến đổi ảnh (Warp Transform)**

* **\_\_four\_point\_transform**
  + Nhận 4 điểm neo (tọa độ [x, y]).
  + Tính ma trận biến đổi phối cảnh để làm phẳng ảnh, đưa ảnh từ dạng nghiêng/chụp về mặt phẳng chuẩn.
* **\_\_two\_point\_transform**
  + Biến đổi dựa trên 2 điểm với chiều ngang cố định.
  + Sử dụng cho các vùng nhỏ như mã đề, số báo danh, hoặc phần trắc nghiệm riêng lẻ.

**Mục đích:** Cố định hình ảnh để nhận diện các vòng tròn và câu trả lời chính xác hơn.

**5. Nhận diện và trích xuất dữ liệu**

* **Nhận diện các contour hình học**
  + **\_\_get\_contours\_circle**
    - Nhận diện các vòng tròn bằng cách tìm các contour có đủ điểm (≥8).
    - Lọc theo bán kính và số lượng pixel trắng (nếu cần).
    - Trả về danh sách tọa độ tâm của các vòng tròn.
  + **\_\_group\_avg\_values**
    - Gom nhóm các tọa độ gần nhau (theo khoảng cách offset).
    - Mục đích: Khắc phục lỗi phát hiện trùng lặp hoặc bỏ sót vòng tròn.
* **Xử lý câu trả lời và mã đề**
  + **\_\_matrix2ans**
    - Chuyển đổi ma trận điểm ảnh thành câu trả lời (A, B, C, D) dựa trên ngưỡng độ sáng (threshold).
    - **\_\_decode**: Biến đổi số (0-3) thành đáp án (A-D).
  + **\_\_get\_info**
    - Phân tích thông tin người làm bài (mã đề, số báo danh).
    - Tìm các vòng tròn tương ứng trong phần mã đề hoặc số báo danh.

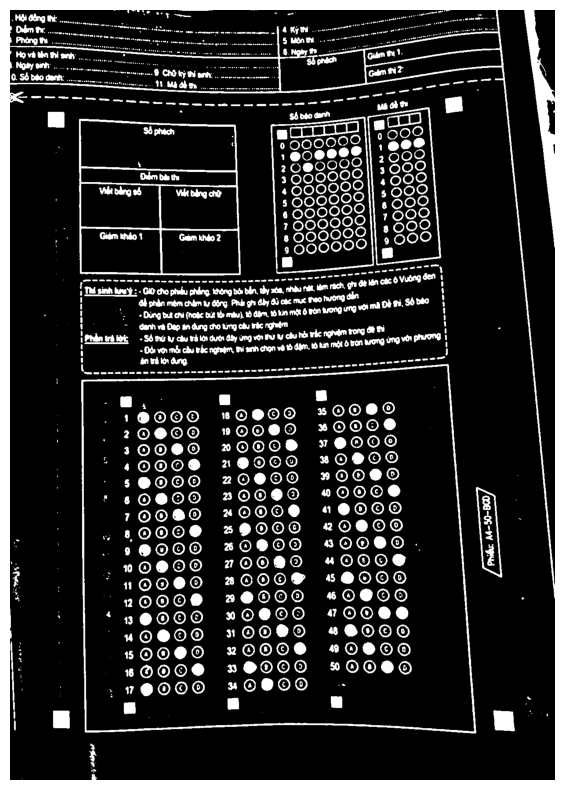
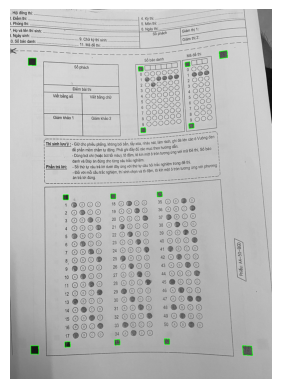
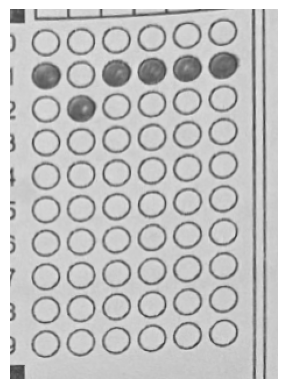
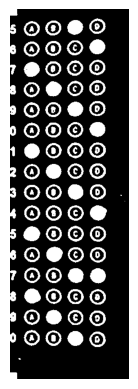
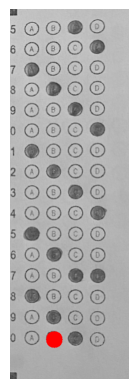
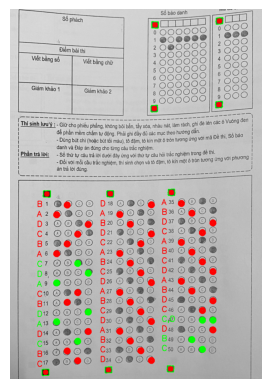
**6. Chấm điểm và xác định kết quả**

* **\_\_find\_correct\_ans**
  + So sánh câu trả lời của học sinh với đáp án đúng (test\_answers).
  + Trả về danh sách câu trả lời đúng.
* **grade**
  + Tổng hợp tất cả các bước:
    1. Phát hiện và biến đổi ảnh về dạng chuẩn.
    2. Phân vùng ảnh: Mã đề, số báo danh, và các câu trả lời.
    3. Nhận diện câu trả lời học sinh.
    4. So sánh với đáp án đúng để tính điểm.
    5. Hiển thị kết quả đúng/sai lên ảnh.

**7. Hiển thị kết quả**

* **Mã màu trên ảnh**
  + Câu đúng: Xanh lá cây.
  + Câu sai: Đỏ.
* Kết quả được hiển thị trực tiếp trên ảnh đã xử lý, dễ dàng đối chiếu với đáp án.

**Sơ đồ khối :**

1. **Nhập dữ liệu (Input Data):**
   * Nhập ảnh bài thi (file ảnh scan).
   * 
   * Nhập đáp án đúng (dưới dạng dictionary).
2. **Tiền xử lý ảnh (Preprocessing Images):**
   * Áp dụng các bộ lọc (Gaussian Blur, Median Blur).
   * Chuyển đổi ảnh sang dạng nhị phân (Binary Threshold).
   * Xử lý để làm nổi bật đường viền và cạnh (Edge Detection).
   * 
3. **Phát hiện điểm mốc (Anchor Point Detection):**
   * Tìm và sắp xếp các đường viền chính.
   * Xác định các điểm mốc (Anchor Points) để hiệu chỉnh ảnh:
     + 4 điểm mốc bên ngoài (chuẩn bị warp ảnh).
     + Điểm mốc vùng mã đề và vùng đáp án trắc nghiệm.
     + 
4. **Biến đổi phối cảnh (Perspective Transformation):**
   * Sử dụng các điểm mốc để:
     + Hiệu chỉnh lại ảnh tổng quan.
     + Cắt vùng ảnh cần thiết: mã đề, thông tin người thi, các câu trắc nghiệm.
     + 
5. **Nhận diện đáp án (Answer Detection):**
   * Phân tích các vùng đáp án (detect circle contours).
   * 
   * Đếm pixel trắng trong mỗi vùng để xác định câu trả lời được chọn.
   * Chuyển đổi dữ liệu ma trận thành các lựa chọn ABCD.
6. **Chấm điểm (Grading and Scoring):**
   * So sánh câu trả lời với đáp án đúng.
   * 
   * Tính điểm dựa trên tỷ lệ số câu trả lời đúng và tổng số câu.
7. **Xuất kết quả (Output Results):**
   * Hiển thị số điểm đạt được.
   * 
   * Tô màu câu trả lời đúng (xanh) và sai (đỏ) trên ảnh.
   * Xuất thông tin người thi và kết quả bài làm.