CYBORG OPENCV TASK - 2

28th May 2025

The goal of this task is to use Python’s OpenCV package to create various functions with following goals:

- Detect colour of the shapes.
- Detect proper coordinates.

Problem Statement:

Little Aryan, a 10-year-old chess prodigy, was preparing for his first national chess championship. While painting with watercolours one afternoon, he accidentally knocked over his paint set onto his prized tournament chessboard. Several squares now appear colored in ***red, green, blue, orange*** completely disrupting the elegant black-and-white pattern.

He wants to identify every square that’s been wrongly colored to repaint them and restore the board. Y***our job is to help Aryan automatically detect these “faulty” squares using Python and OpenCV.***

Each image in the ***Test_images*** folder of Task_2 contains a ***8x8*** chessboard containing ***64 squares.***

A sample image can be found below for reference:

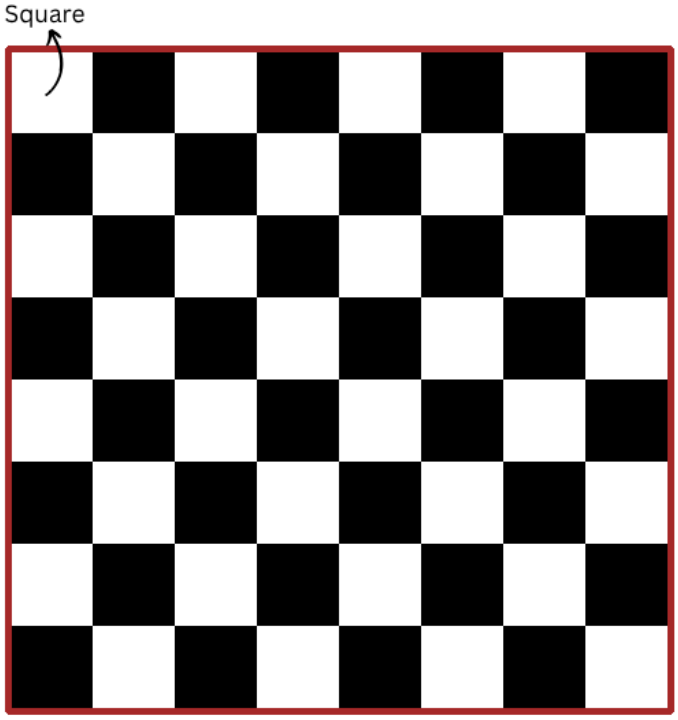
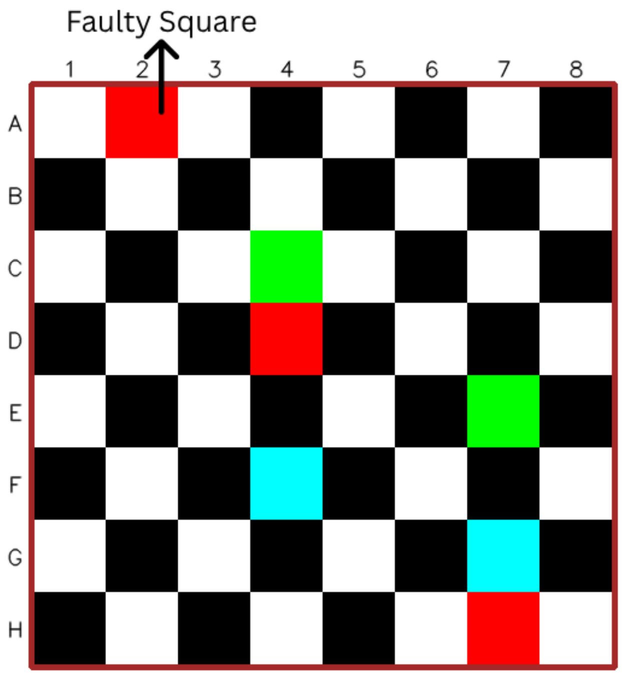


Image Description:



- Clone the repo: GitHub in your local system.

Directory Structure:

- The task directory (Task_2A/) contains:
  - evaluator
  - main.py (where you implement your solution).
  - Test_images/ (a folder containing 10 test images: faulty_chessboard_1.png to faulty_chessboard_10.png).

Task Details

- **Chessboard Layout**:
  - The chessboard has 8 rows (labeled 1 to 8) and 8 columns (labeled A to H).
  - A1 is the bottom-left square, A8 is the top-left, H1 is the bottom-right, and H8 is the top-right.
- **Faulty Squares**:
  - Some squares have incorrect colors (Red, Green, Blue, or Orange) instead of Black or White.

- You need to identify these squares by sampling the color at the center of each square.

Output

- Implement the function detect_faulty_squares(input_image) in main.py.

(For more info refer the function in main.py)

Submission:

- **Testing**:
  - Use test.py to test your implementation on faulty_chessboard_1.png.

- **Evaluator**:
  - Run the evaluator binary to test your implementation on all 10 images:
  - Put the command *./evaluator* inside the Task_2A directory.
  - If any test cases pass, it will generate a submission file (task_2A_[roll_number].encrypt).
  - Put the file inside the Task_2A directory.

- Make your own repo and push all the changes to it.
- Make sure to keep it **private.**

BONUS TASK:

Objective:

You are provided with a distorted chessboard image that exhibits fisheye lens distortion. The chessboard is an 8x8 grid with five ArUco markers placed at specific positions. Your task is to correct the fisheye distortion, apply a perspective transform to rectify the chessboard into a top-down view, and detect the ArUco markers. Finally, print the IDs of the ArUco markers in a serial order from top to bottom.

Input

- **Image**: A single distorted chessboard image provided as a NumPy array (loaded using cv2.imread()).
  - The chessboard suffers from fisheye lens distortion.
  - Five ArUco markers are placed on the chessboard at unknown positions.
  - The image is located in the Task_2B_BONUS/ directory as distorted_chessboard.png.

Directory Structure:

- The task directory (Task_2B_BONUS/) contains:
  - main_bonus.py (where you implement your solution).
  - distorted_chessboard.png (the input image).

Steps to Solve:

1. Undistort the Image:
2. Detect ArUco Markers:
3. Print Marker IDs. (No need to print the coordinates, only ids required.)

Submission:

Push the changes to the above repo created by you.

DEADLINE : 2nd June 11:59pm.