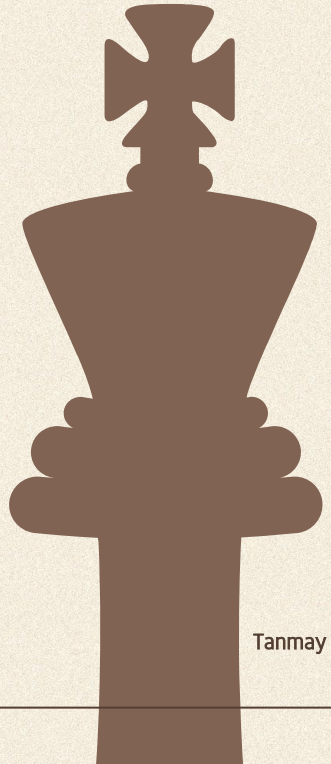
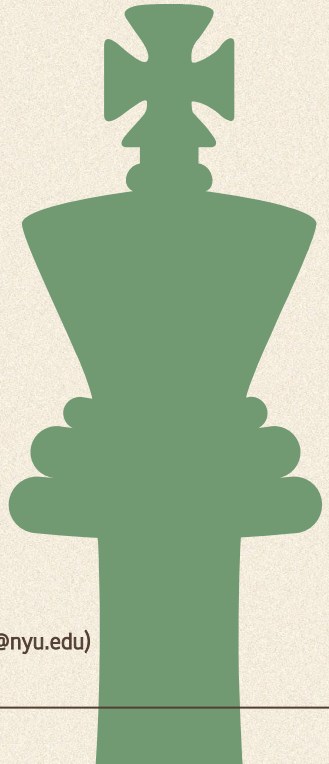


Automated Chess Board Recognition and Interaction

Using a Top-View Setup



Tanmay Rathi (tr2452@nyu.edu) • Simardeep Singh Mehta (sm11377@nyu.edu) • Avadhoot Kulkarni (ak10576@nyu.edu)

Project Motivation

Interest in AI & Games: Application of computer vision in real-time board game tracking.

Academic Challenge: Implementing course-learned techniques (Hough transforms, filtering) in a practical project.

Real-World Problem: Automating chess piece tracking without expensive equipment.

- The data captured by electronic boards is used to broadcast games live over the internet.

Project Motivation

Electronic Chess Boards:

- Many professional chess tournaments use electronic chess boards.
- These boards have sensors under each square that detect the presence and identity of pieces. When a player makes a move, the board automatically records it and transmits the data to a connected computer or an online system.
- The data captured by electronic boards can be used to broadcast games live over the internet.
- Moves recorded during a game can be fed into chess analysis software such as Stockfish or Komodo, which use advanced algorithms to evaluate the positions and suggest improvements or blunders.

Project Motivation

Current Solutions:
High-cost sensor-equipped chess boards or software-dependent on deep learning or template matching.

Limitations of Deep Learning:
- **Resource Intensive:** Requires significant computational power and extensive training data.
- **Complex Setup:** Needs ongoing training and adjustment to maintain accuracy across different chess sets and lighting conditions.

Limitations of Template Matching:
- **Performance Issues:** Can be computationally expensive and slow, especially when matching each piece across multiple potential positions.

Project Motivation

Our Approach:
Using stable camera setup and simple computer vision techniques.

- **Cost-Effective:**
  Minimal hardware requirements, no need for expensive setups.

- **Efficiency:**
  Processes frames quickly, enabling near real-time tracking.
  (~0.5 to 1 second)

```
Elapsed time: 1.3387219905853271 seconds
Elapsed time: 0.991455078125 seconds
Elapsed time: 0.8979179859161377 seconds
Elapsed time: 0.6468210220336914 seconds
Elapsed time: 0.5626230239868164 seconds
```

- **Simplicity and Robustness:**
  Less prone to errors in varied lighting and different chess board or piece appearances.

- **Scalability:**
  Easily adaptable to different environments without extensive reconfiguration.

Current Setup:

Web application that processes images saved in a specific folder

Live Feed Adaptation:

Images from live feed can be saved in the designated folder, *(Allowing for real-time game tracking and analysis)*

Initialization

Load the initial chessboard image.
Click to start the game.

Reset

Clears the board for a new game.

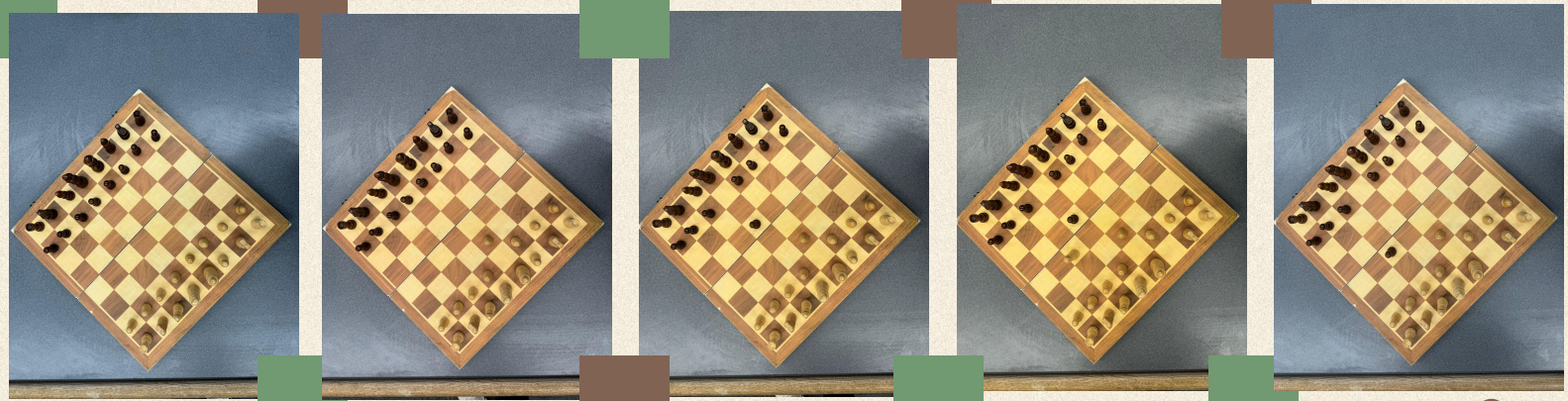
Next Move

Saves the current board image.

Triggers processing of subsequent images, detecting changes by subtracting edge magnitudes from the previous frame.

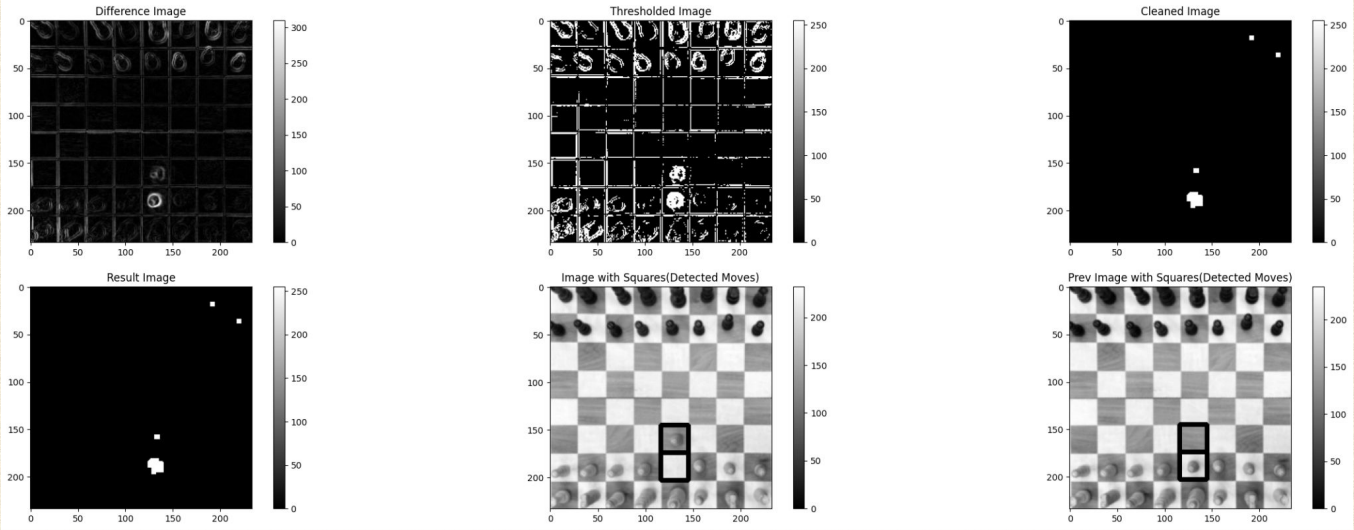
Shows movement of chess pieces on the UI by identifying differences in the board state.

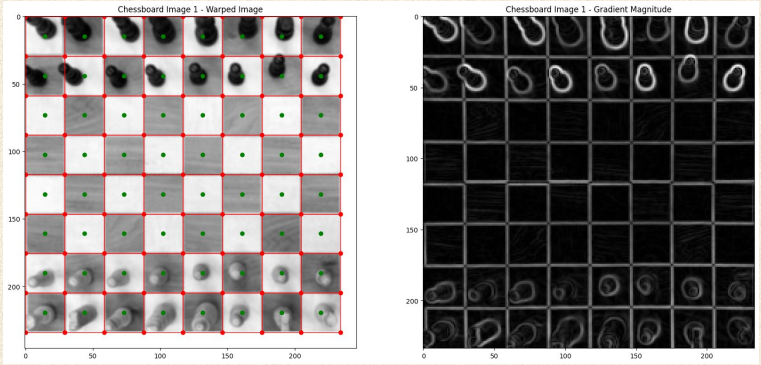
EXTRACTED FRAMES FROM LIVE GAME

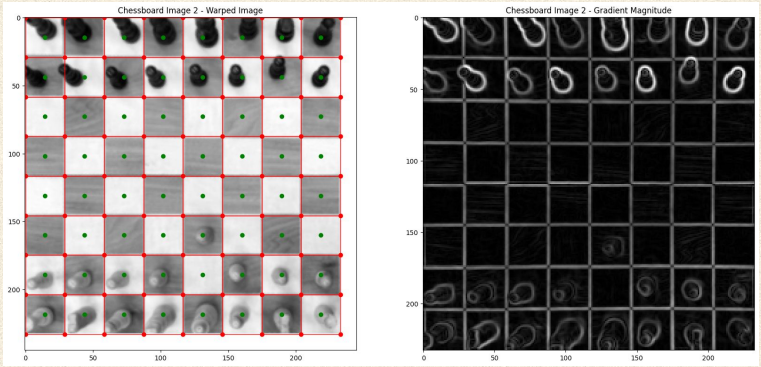


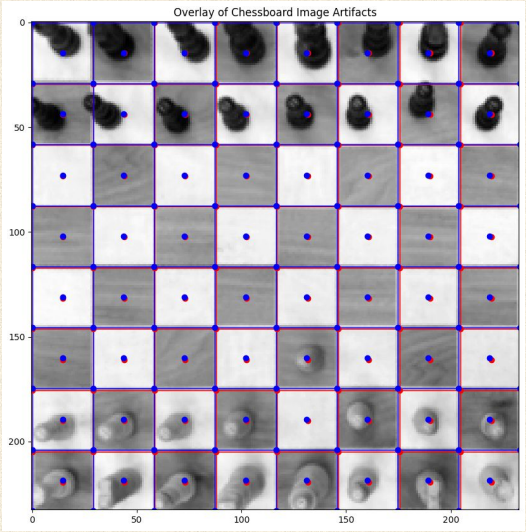
Elapsed time: 1.549391508102417 seconds

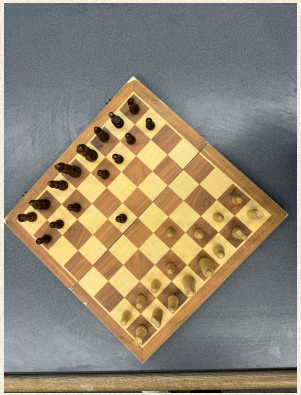
Move: e2 to e3




Chessboard Image 1 - Warped Image
Chessboard Image 1 - Gradient Magnitude

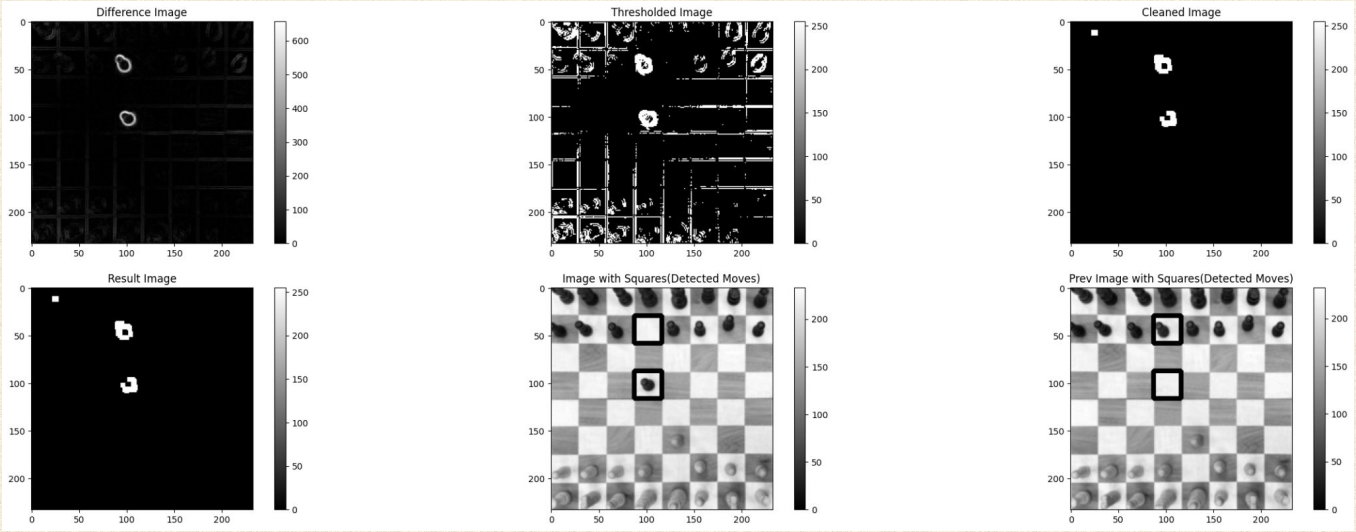

Chessboard Image 2 - Warped Image
Chessboard Image 2 - Gradient Magnitude

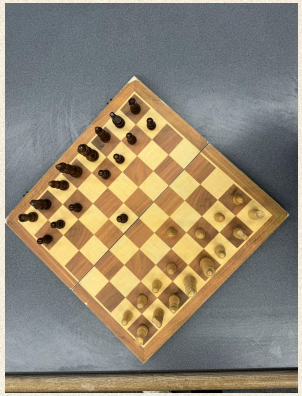

Overlay of Chessboard Image Artifacts



Elapsed time: 1.8433990478515625 seconds

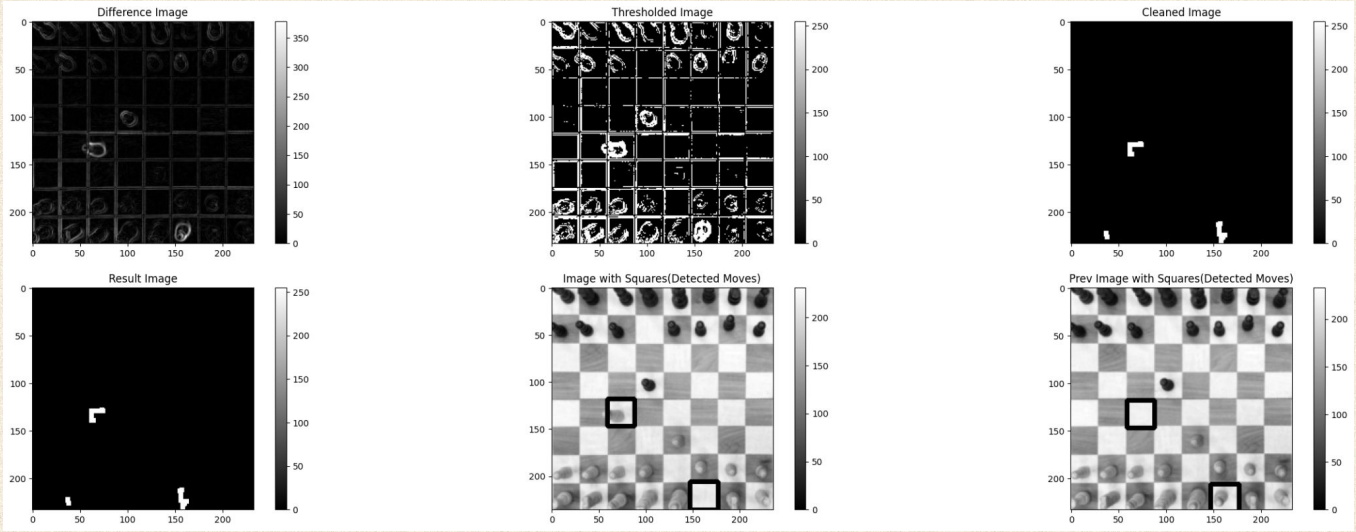
Move: d5 to d7

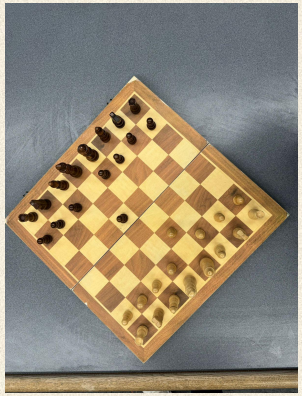


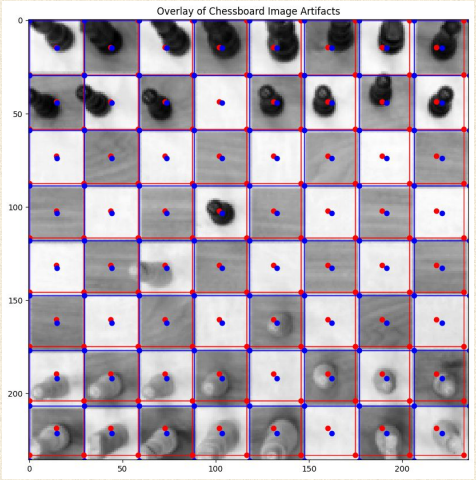


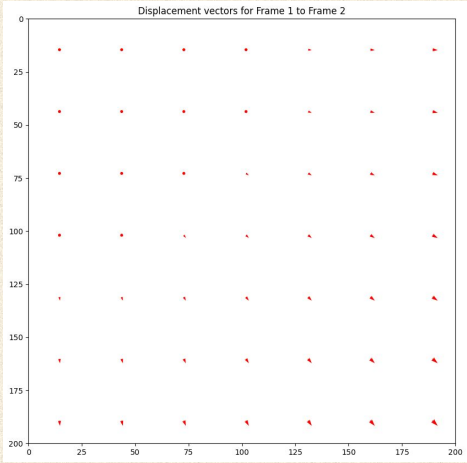
Elapsed time: 1.5658328533172607 seconds

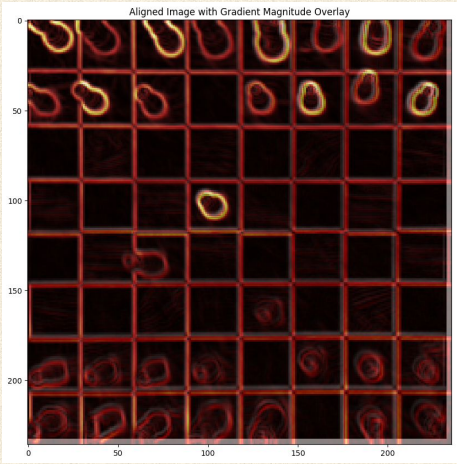
Move: f1 to c4





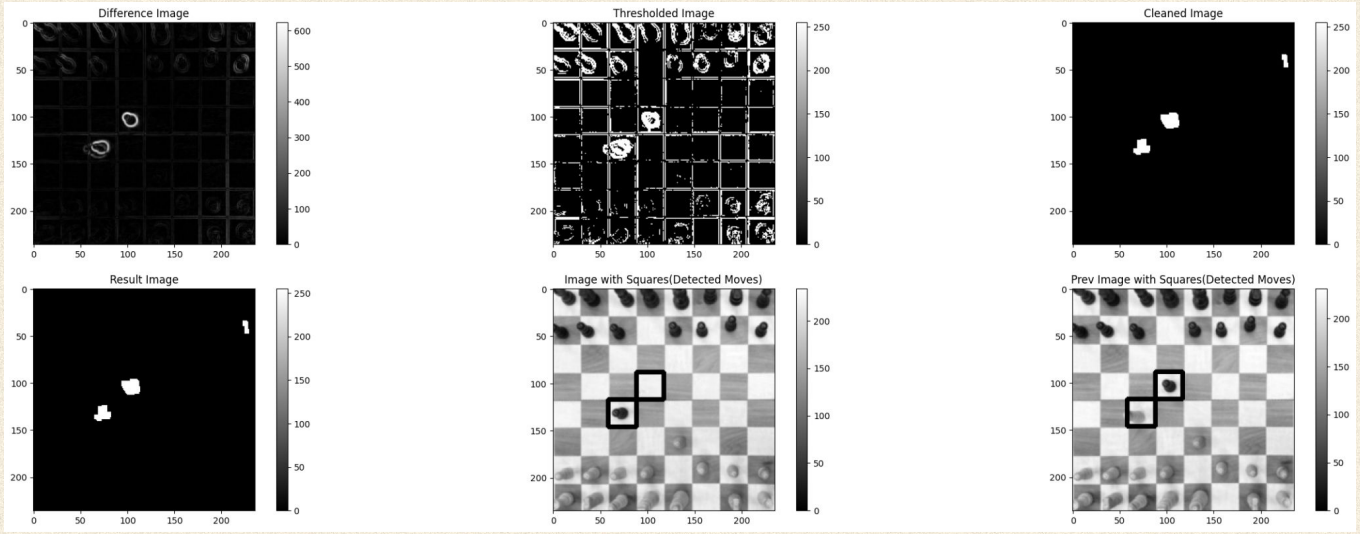

Overlay of Chessboard Image Artifacts
0
50
100
150
200
0
50
100
150
200


Displacement vectors for Frame 1 to Frame 2
0
25
50
75
100
125
150
175
200
0
25
50
75
100
125
150
175
200


Aligned Image with Gradient Magnitude Overlay
0
50
100
150
200
0
50
100
150
200

Elapsed time: 1.4210774898529053 seconds

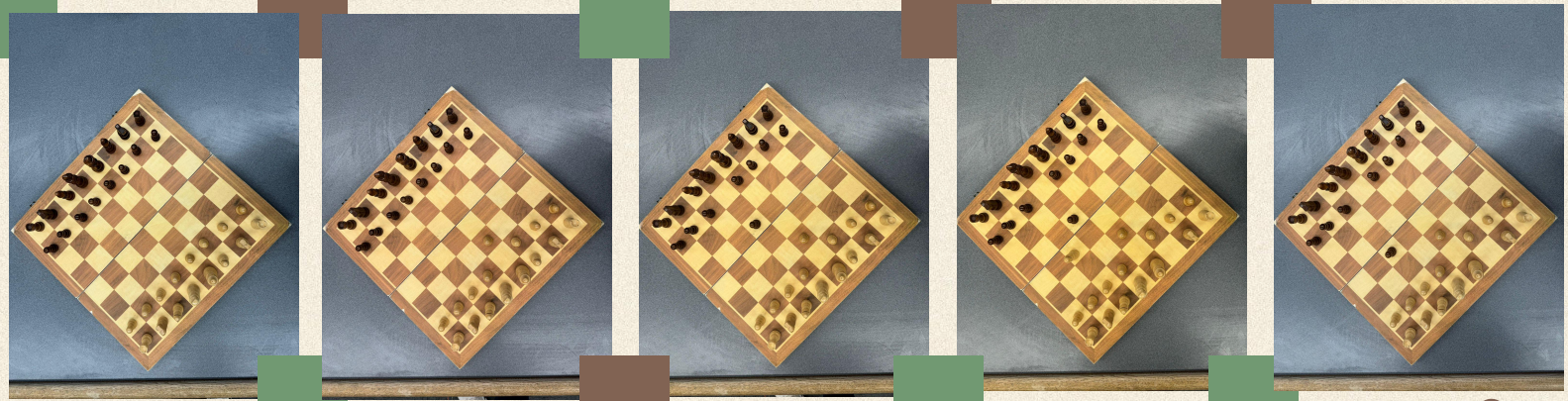
Move: d5 to c4

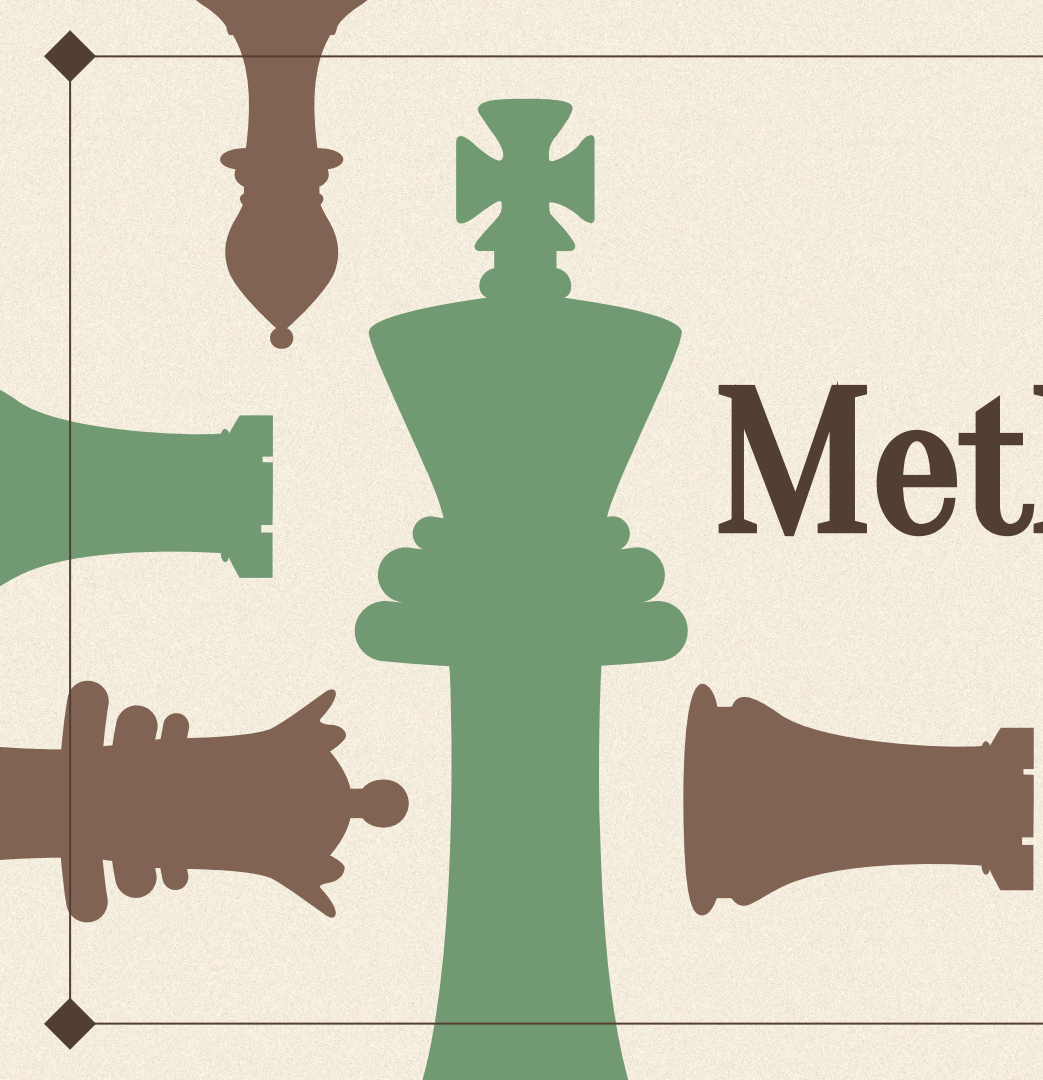


DEMO!

Demonstration of our Automated Chess Board Recognition System

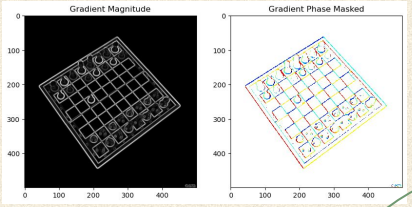
EXTRACTED FRAMES FROM LIVE GAME





Methodology

Lets dig into the magic!



Detect Saddle Points:
Apply 2nd derivative logic to locate saddle points as potential chess square corners.

Input Images

Load two sequential images of the chessboard (before and after a move).

Preprocessing

Normalize images by rotating, resizing, and converting to grayscale for consistent processing.

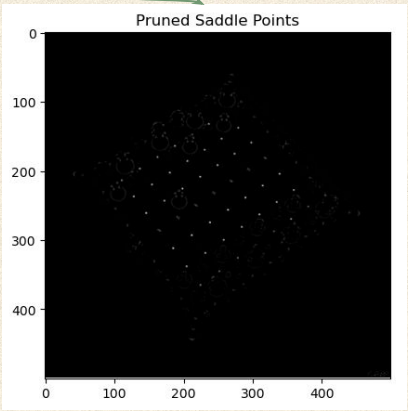
- Rotate as needed
- Resize to max 500x500px
- Convert to grayscale

Saddle Point Detection

Apply Sobel operators to compute gradients and detect saddle points for feature localization.

- Compute x and y gradients (Sobel)
- Calculate magnitudes and phases
- Detect saddle points via derivatives

Prune based on intensity
to isolate prominent saddle points.
(based on a dynamic threshold)



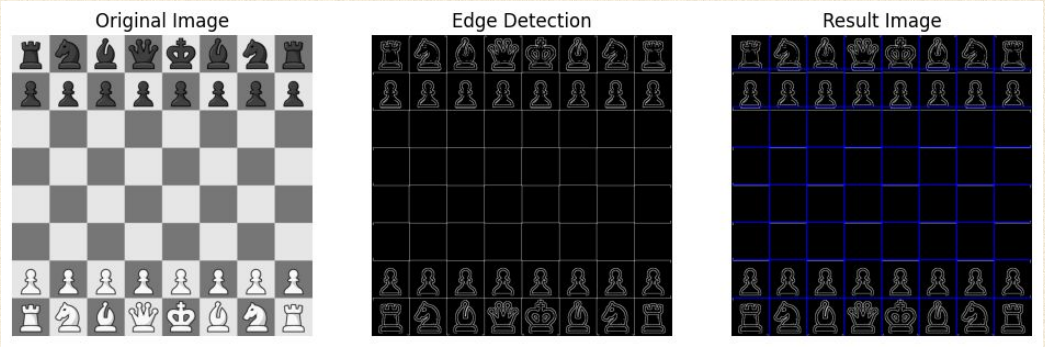
Results on 2D Images:

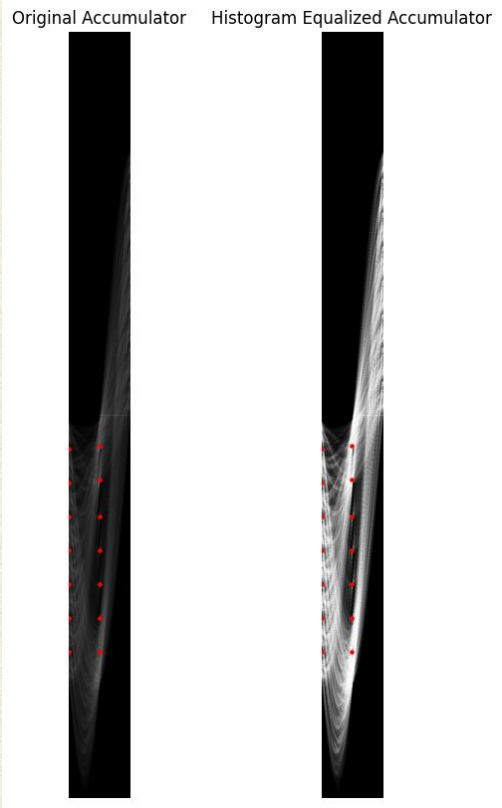
We had explored the use of Hough Transform for line detection as an alternative to our primary method of detecting saddle points.

We employed the Hough Line detection with **complex angle logic and distance filtering** to **enhance accuracy.**

Angle Calibration: Configuring the Hough Transform to detect lines at specific angles corresponding to the expected orientations of the chess grid.

Distance Thresholds: Implementing a minimum distance parameter to ensure that only the most prominent lines, representing the chess grid, are detected and minor variations or noise are disregarded.

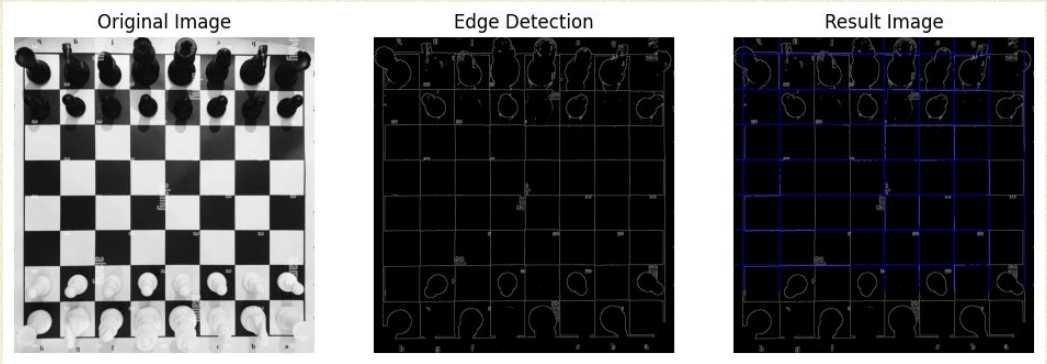


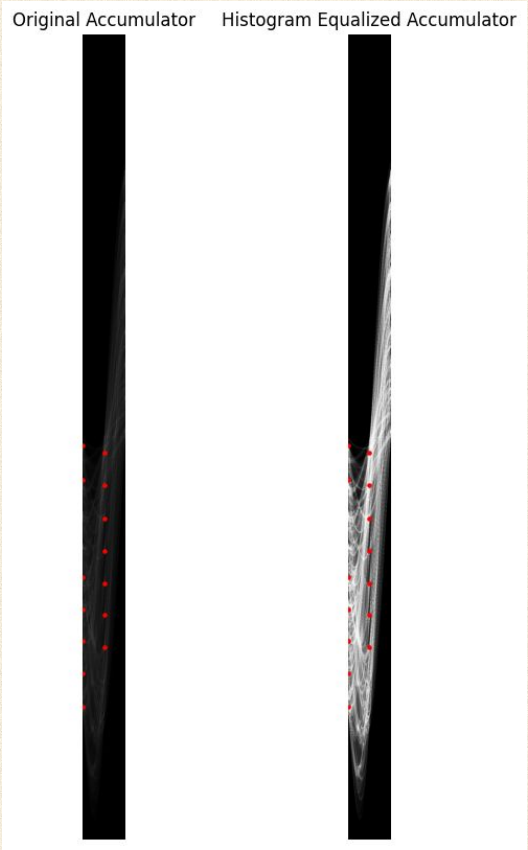


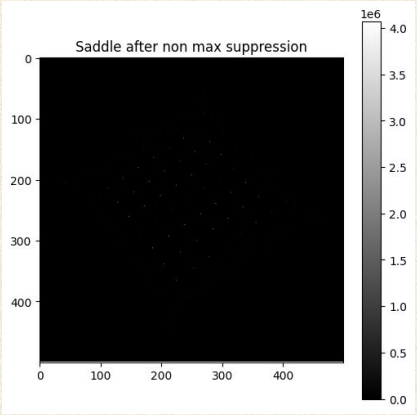
Limitations of Hough Lines in Complex Imaging Conditions:

Occlusions Caused by Chess Pieces: Often, the presence of chess pieces disrupts the continuity of the grid lines, leading to incomplete or inaccurate line detection.

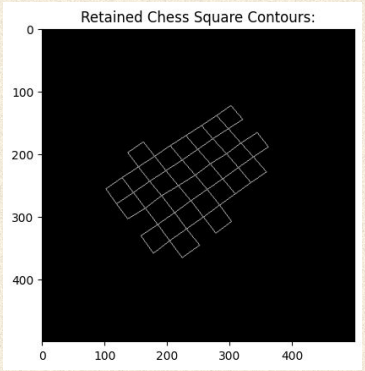
Variable Lighting and Shadows: These elements introduce additional complexity, as they can create false positives or obscure real lines.







Apply non-maximum suppression to the saddle points to isolate the most prominent ones



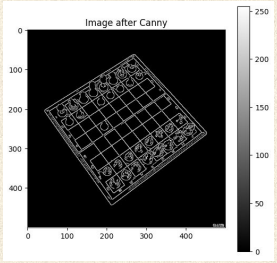
Contour Extraction and Pruning

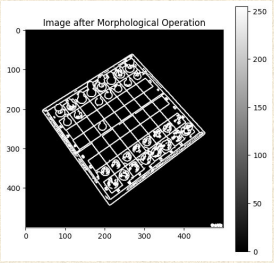
Contour Identification:
Use Canny edge detection to outline sharp changes.

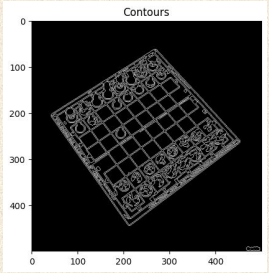
Extract contours enhanced by morphological operations.

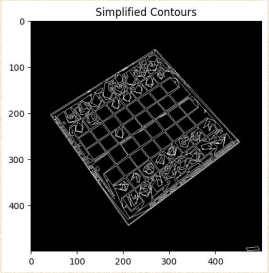
Simplify Contours:
Apply Douglas-Peucker algorithm to reduce contour complexity.

Retain Chess Square Contours:
Filter contours to retain only those approximating square shapes, crucial for identifying chessboard grid.









Homography and Grid Alignment

Result:
A consistently aligned chessboard that facilitates precise detection & analysis of chess piece movements.

Homography Calculation

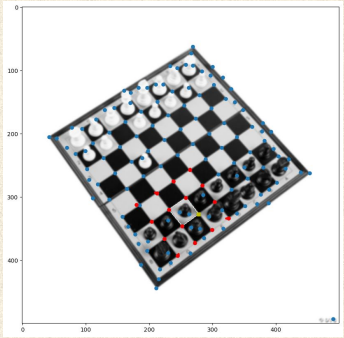
Correct perspective distortions to achieve a top-down view of the chessboard.

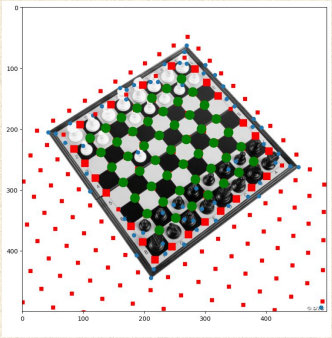
- Identify corner points or contours that outline the chessboard.
- Compute the homography matrix using cv2.getPerspectiveTransform().
- Apply the transformation with cv2.warpPerspective() to align the image.

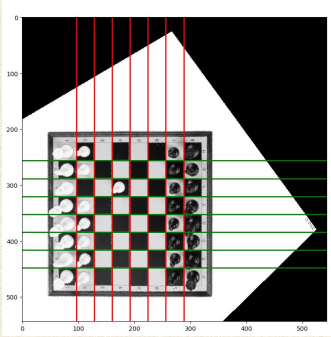
Grid Refinement

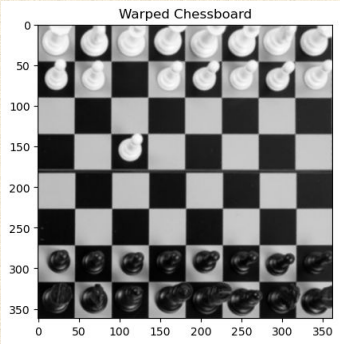
Ensure the grid is accurately mapped to enhance move detection accuracy

.

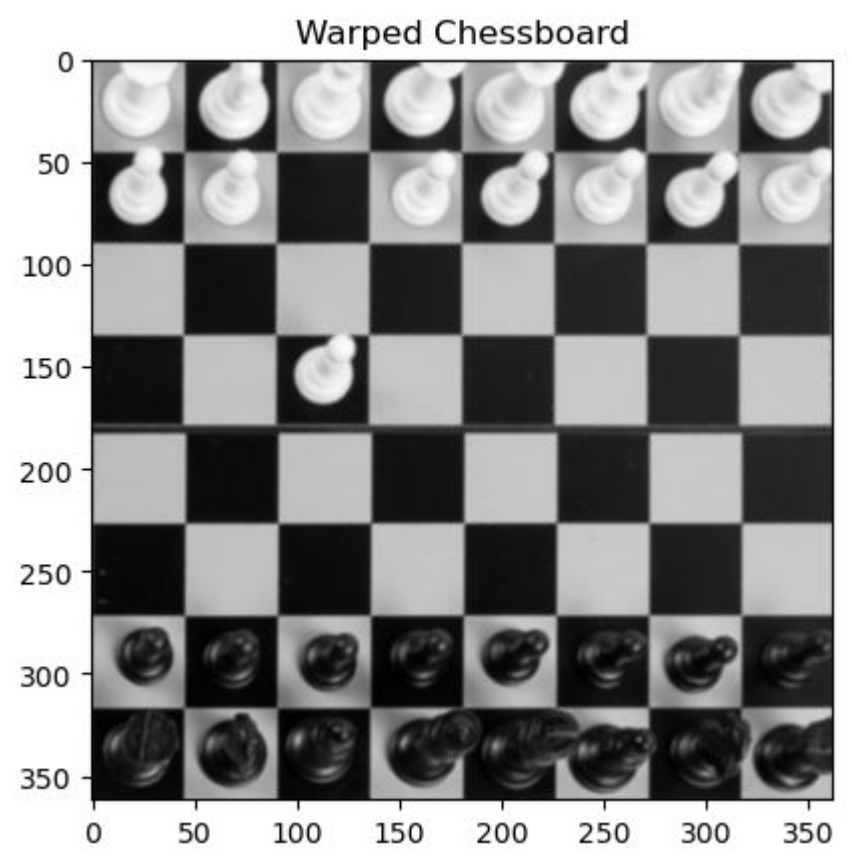


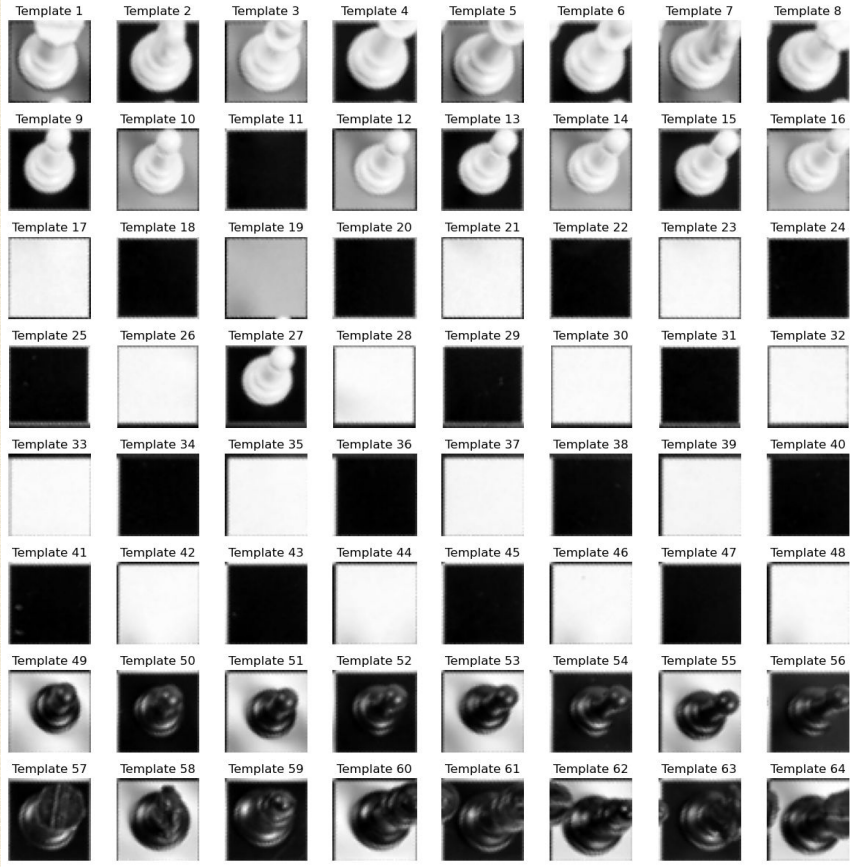


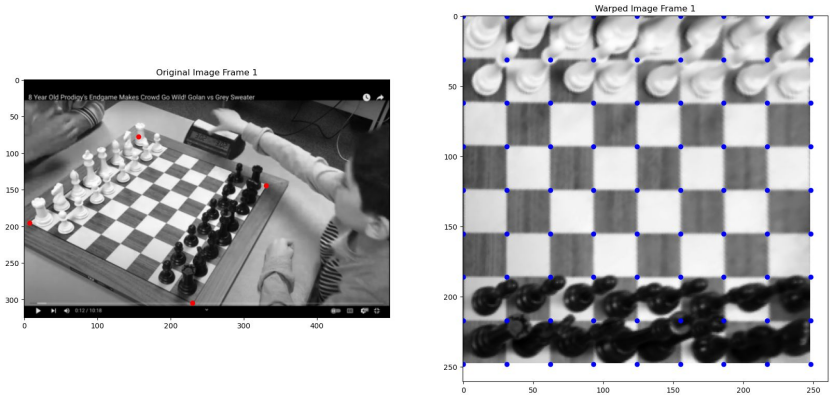


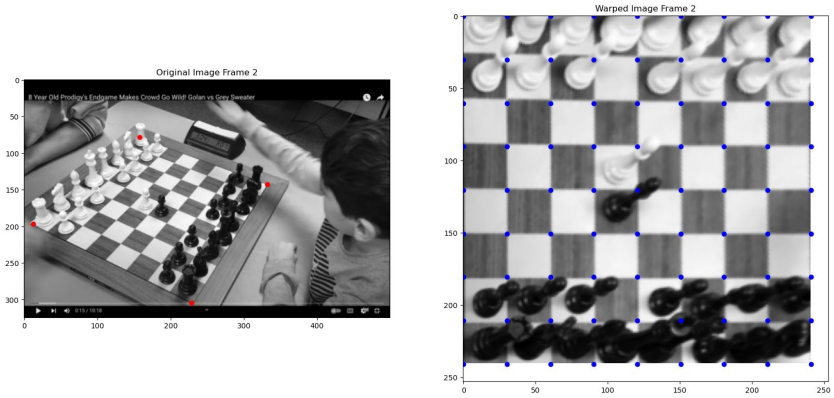


CHANGE DETECTION


Warped Chessboard
0
50
100
150
200
250
300
350
0
50
100
150
200
250
300
350


Template 1
Template 2
Template 3
Template 4
Template 5
Template 6
Template 7
Template 8
Template 9
Template 10
Template 11
Template 12
Template 13
Template 14
Template 15
Template 16
Template 17
Template 18
Template 19
Template 20
Template 21
Template 22
Template 23
Template 24
Template 25
Template 26
Template 27
Template 28
Template 29
Template 30
Template 31
Template 32
Template 33
Template 34
Template 35
Template 36
Template 37
Template 38
Template 39
Template 40
Template 41
Template 42
Template 43
Template 44
Template 45
Template 46
Template 47
Template 48
Template 49
Template 50
Template 51
Template 52
Template 53
Template 54
Template 55
Template 56
Template 57
Template 58
Template 59
Template 60
Template 61
Template 62
Template 63
Template 64



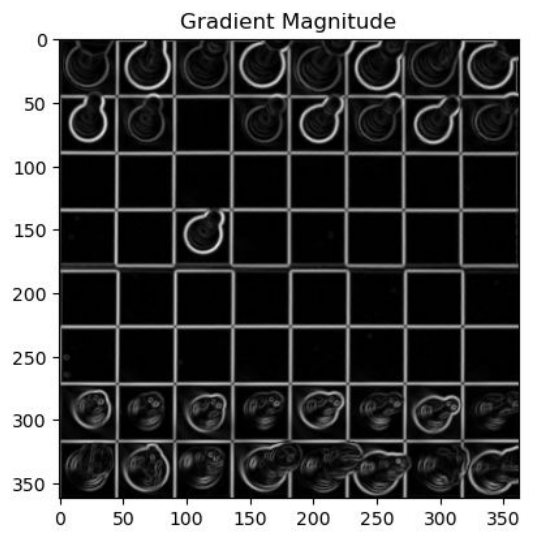


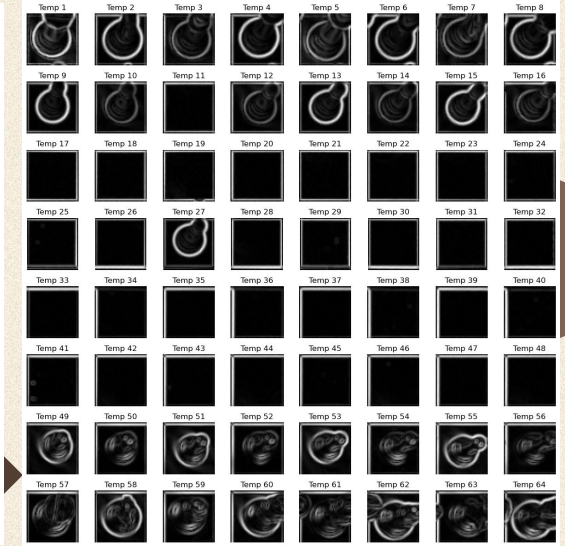
Potential Use: For non-top-view angles

Methods like ***deep learning or template matching*** could leverage the generated templates for more precise piece recognition.

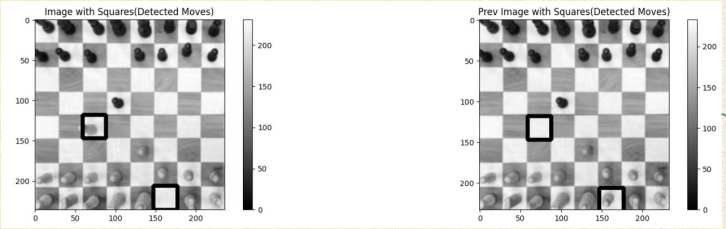
Chosen Method: Image Subtraction

Selected for its **real-time** processing capability and **lower computational cost.**





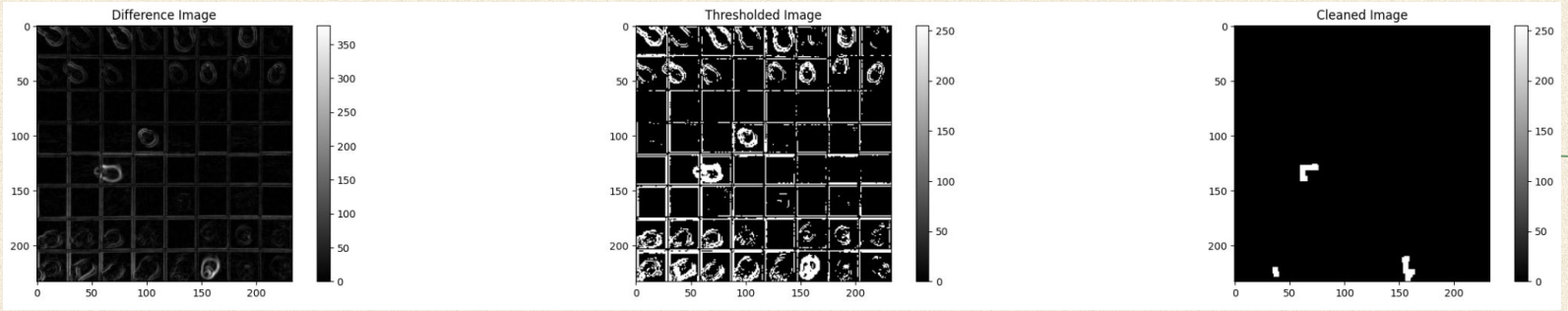
Change Detection



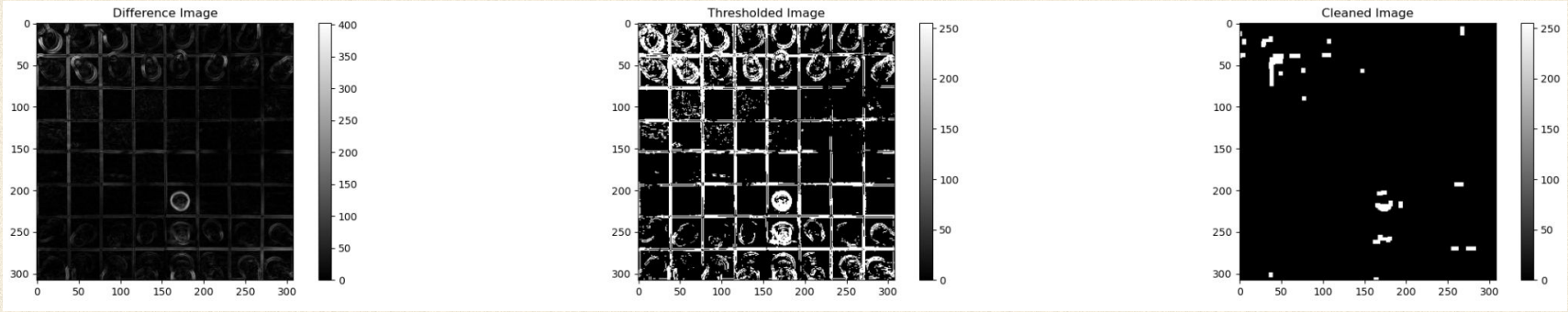
1. Subtracting Images:

Identify areas of change between two aligned chessboard images.

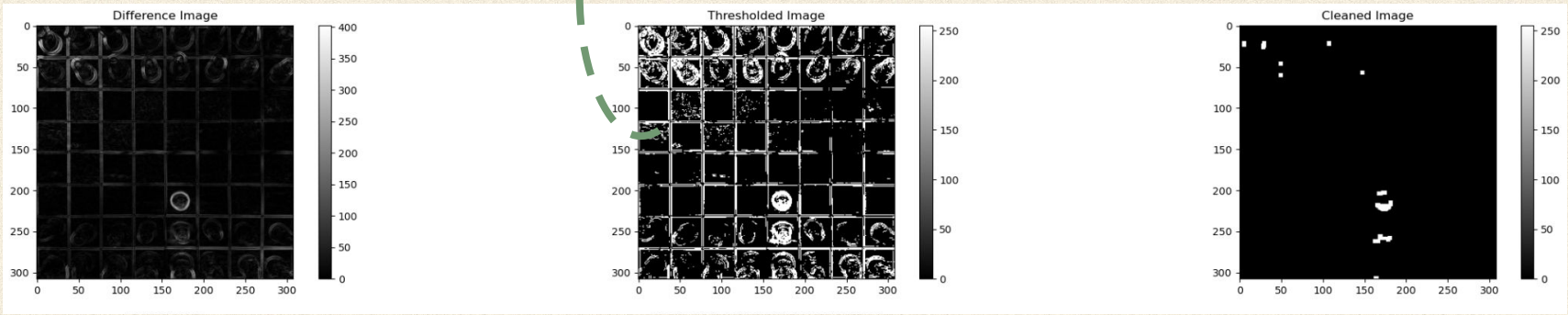
- ***Alignment:*** Use homography to ensure both images are perfectly aligned.
- ***Subtraction:*** Calculate the absolute difference between the gradient magnitude images of consecutive frames.



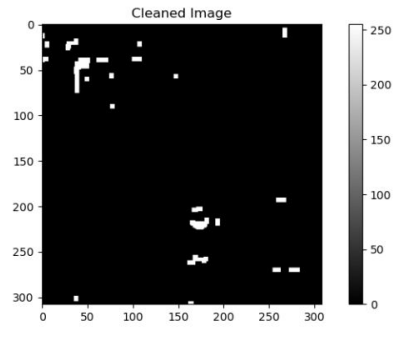
Line Removal Logic (Thresholding)

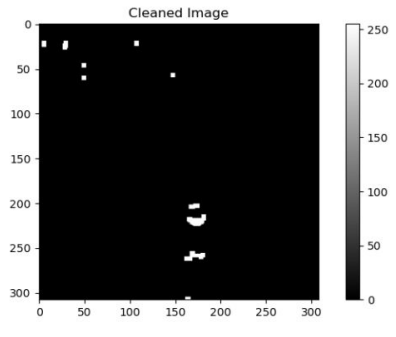


Grid Line Pixels set to 0



Line Removal Logic (Thresholding)





Purpose

Prevent grid lines from interfering with piece movement detection.

Action

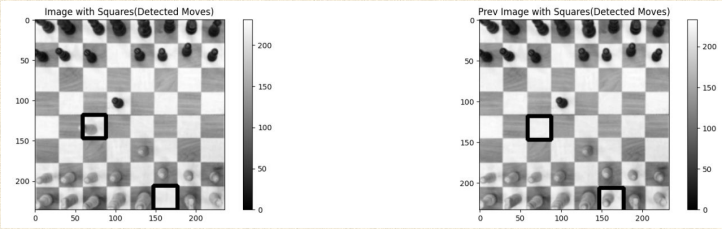
Remove Lines: Erase horizontal and vertical lines between grid points on the chessboard.

Benefits

Enhanced Clarity: Improves visual clarity for subsequent processing steps.

Increased Accuracy: Reduces false detections caused by grid lines in the analysis.

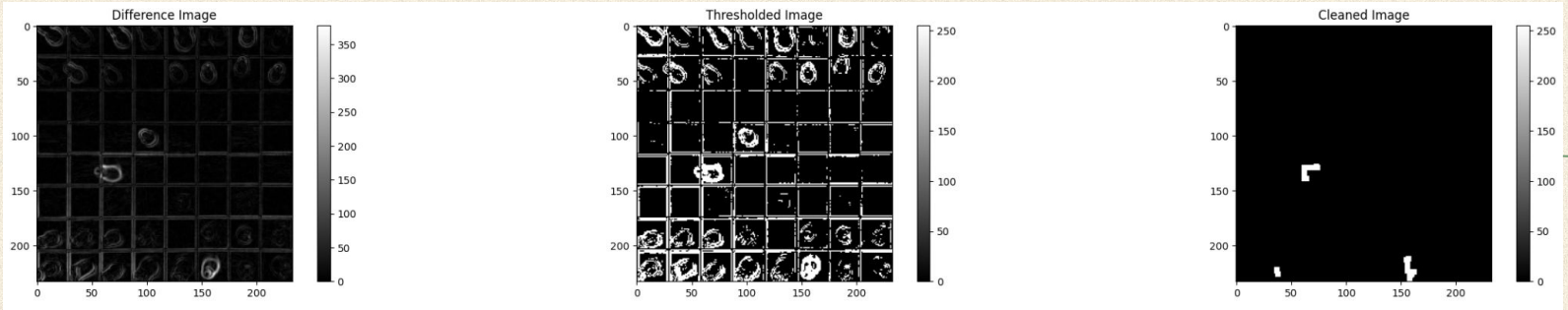
Change Detection



2. Thresholding and Cleaning:

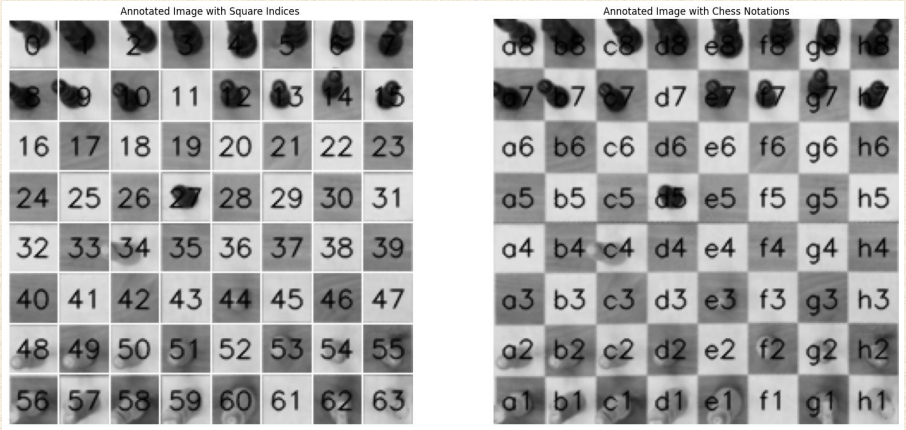
Isolate significant movements and reduce noise.

- ***Thresholding:*** *Apply a binary threshold to the difference image to filter out minor changes*
- ***Morphological Operations:*** *Use opening and closing operations with a 5x5 kernel to clean the thresholded image.*
  Removes spurious noise and defines the changes more clearly.



Move Detection and Notation

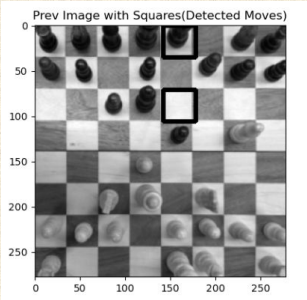
Result: The system successfully converts the analyzed image data into standard chess notations, such as "e2 to e4".

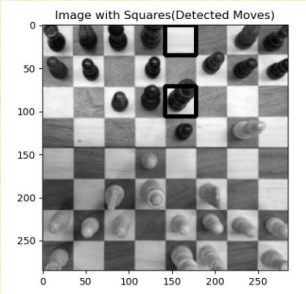


Analyzing Detected Changes

Convert detected image changes into specific chess moves.

1. **Identify Top Changes:** Select the two areas with the highest change intensities from the cleaned image.
2. **Determine Move Direction:** Analyze the relative intensity and location to ascertain the direction of the move (from start square to end square).





Chess Notation Conversion

Translate grid indices into standard chess notation for clarity and game tracking.

Convert the grid indices of the start and end points into chess notations (e.g., e2 to e4).

Automated Chess Board Recognition and Interaction Using a Top-View Setup

Detected Move:
Displays the annotated move on the chessboard.

Processing Time:
Shows the quick response time.

User-Friendly Interface:
Easy to use and intuitive.

Summary

01 Effective Strategy

Demonstrated the effectiveness of using image subtraction for real-time chess move detection.

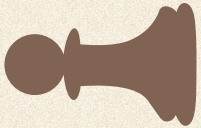
02 Practical Application

Showcased a practical application that enhances digital chess interfaces without the need for expensive hardware.

03 Technical Achievements

Successfully implemented and refined techniques such as homography alignment, grid line removal, and dynamic thresholding.

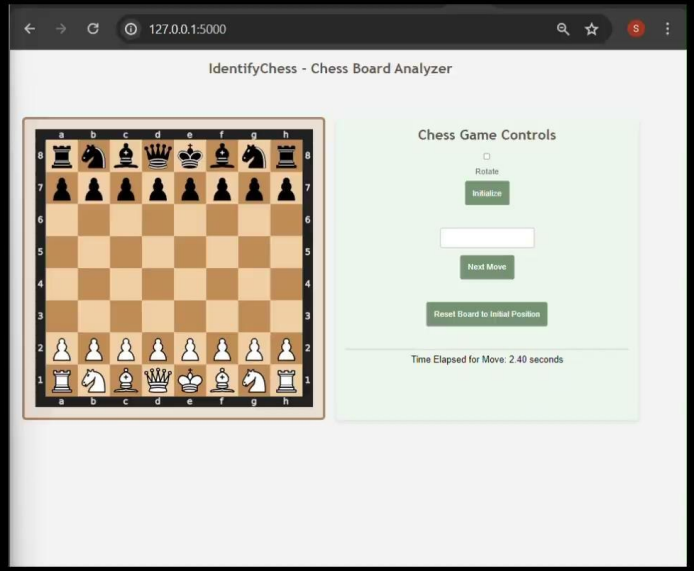
Future Work



Advanced Detection Algorithms (CNNs/Template Matching)

Multi-Angle Support

User Interface Improvements


127.0.0.1:5000
IdentifyChess - Chess Board Analyzer
Chess Game Controls
Rotate
Initialize
Next Move
Reset Board to Initial Position
Time Elapsed for Move: 2.40 seconds

THANK YOU!

OUR TEAM

Simardeep Singh Mehta

sm11377@nyu.edu

Avadhoot Kulkarni

ak10576@nyu.edu

Tanmay Anil Rathi

tr2452@nyu.edu