



Digital Image



Lovecat888

Final Project:
**Chess
Detection**



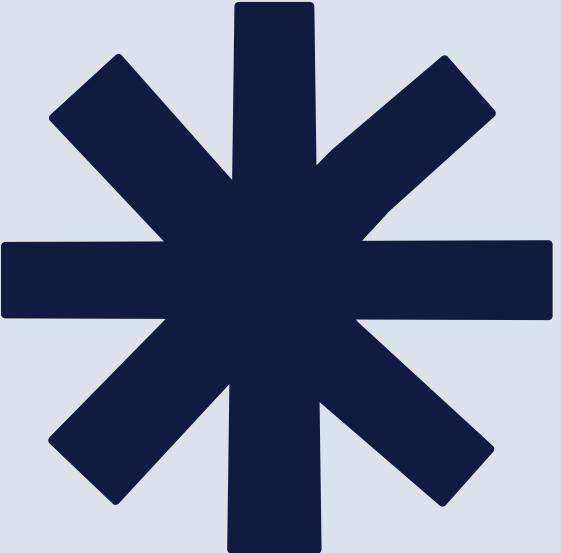


Content

Pipeline + Detail Methodology

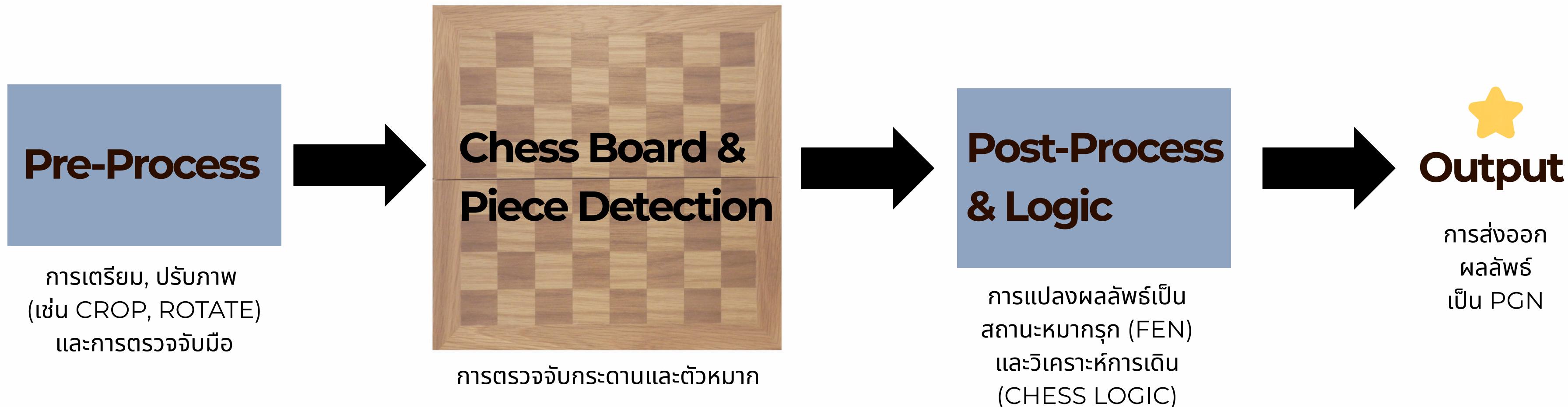
Evaluation

Role and Responsibilities





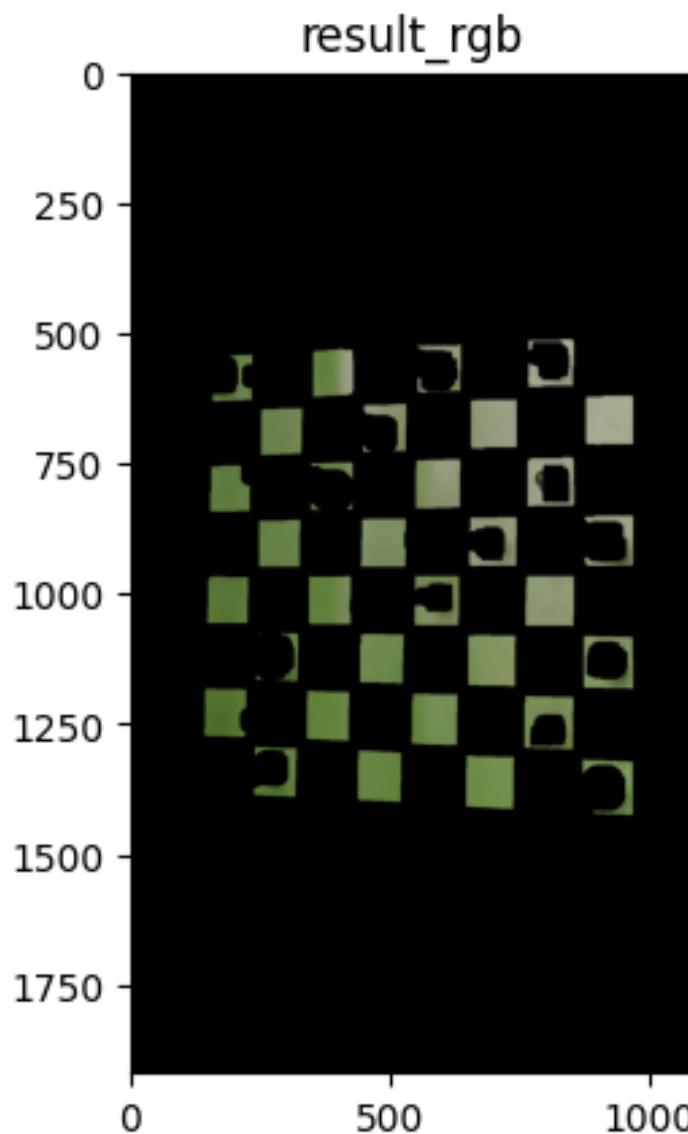
Pipeline





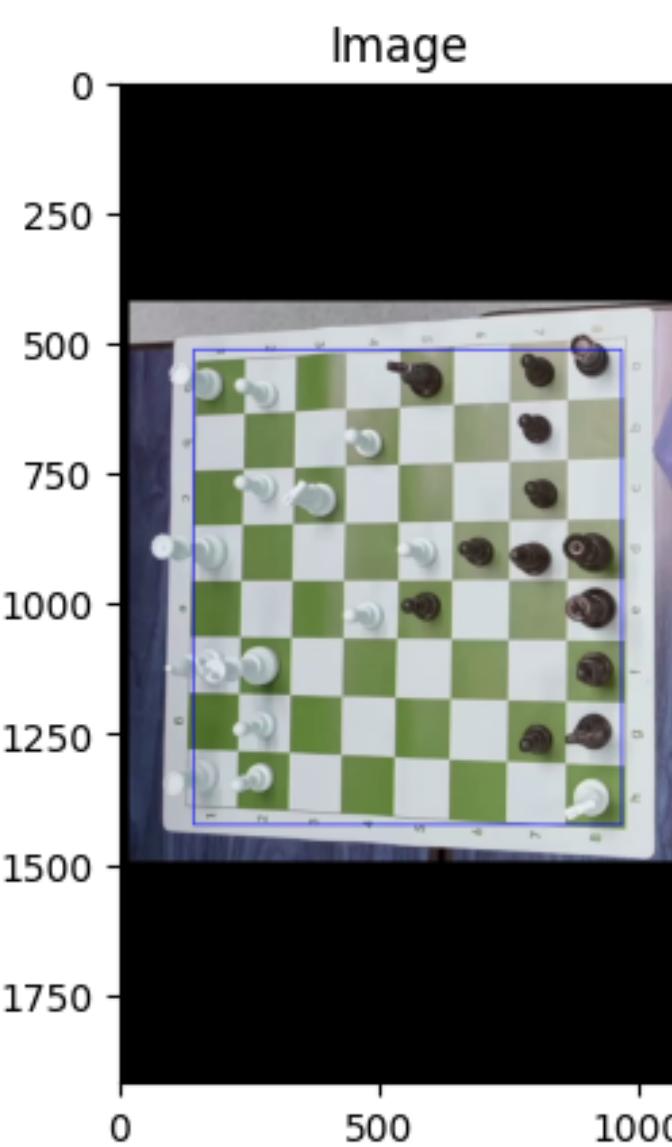
Pre-Process

getGreenSquare()



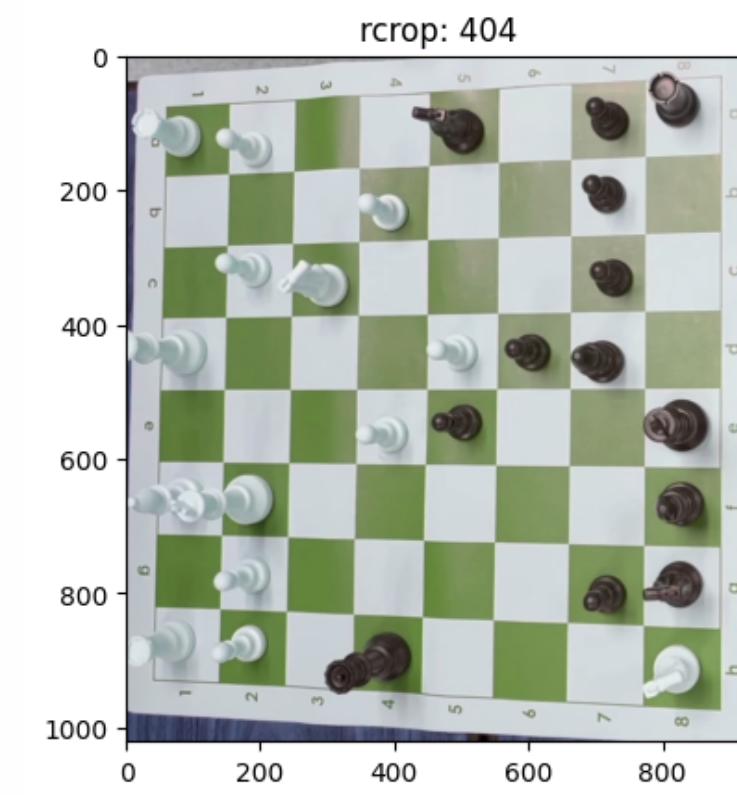
สร้าง Mask สีเขียว
เพื่อแยกส่วนกรอบ

GetCorner() และ
CropAndPadding()



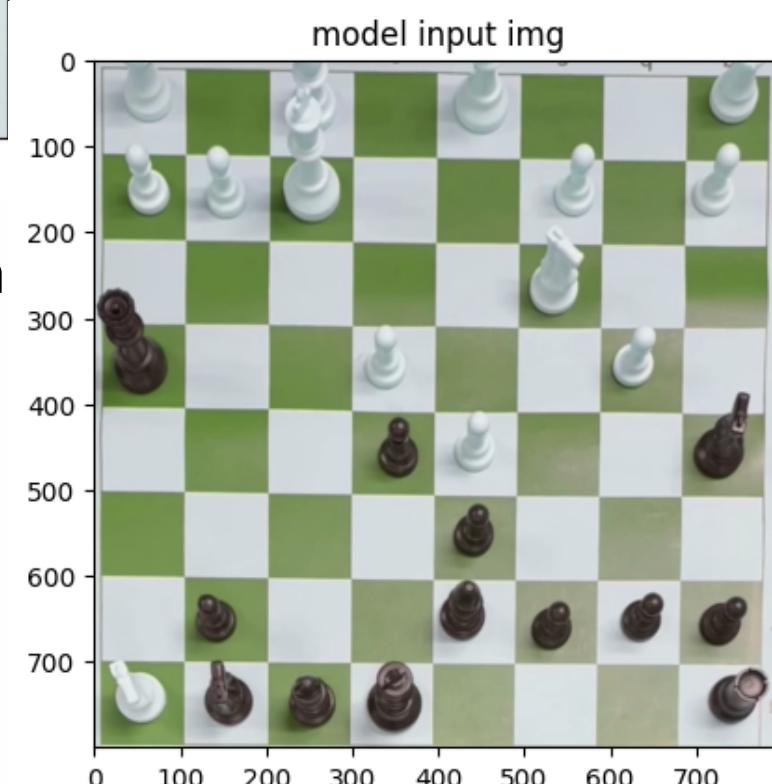
- ใช้ภาพ Mask เพื่อหาค่า Min & Max ของ x & y
- ทำการ Crop ภาพ RGB ตามขอบเขตที่หาได้ พร้อม
เพิ่ม Padding เพื่อให้ได้ภาพกรอบที่ซัดเจน

rotateImageBasedOn
BlackPiece()



บันคะแบบตามกิศ เช็คชิบกี่หมากด้อยู่มาก
สุด (บน, ล่าง, ซ้าย, หรือ ขวา) เพื่อหมุนให้
กระดานถูกจัดวางในกิศทางมาตรฐาน

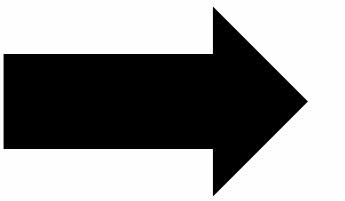
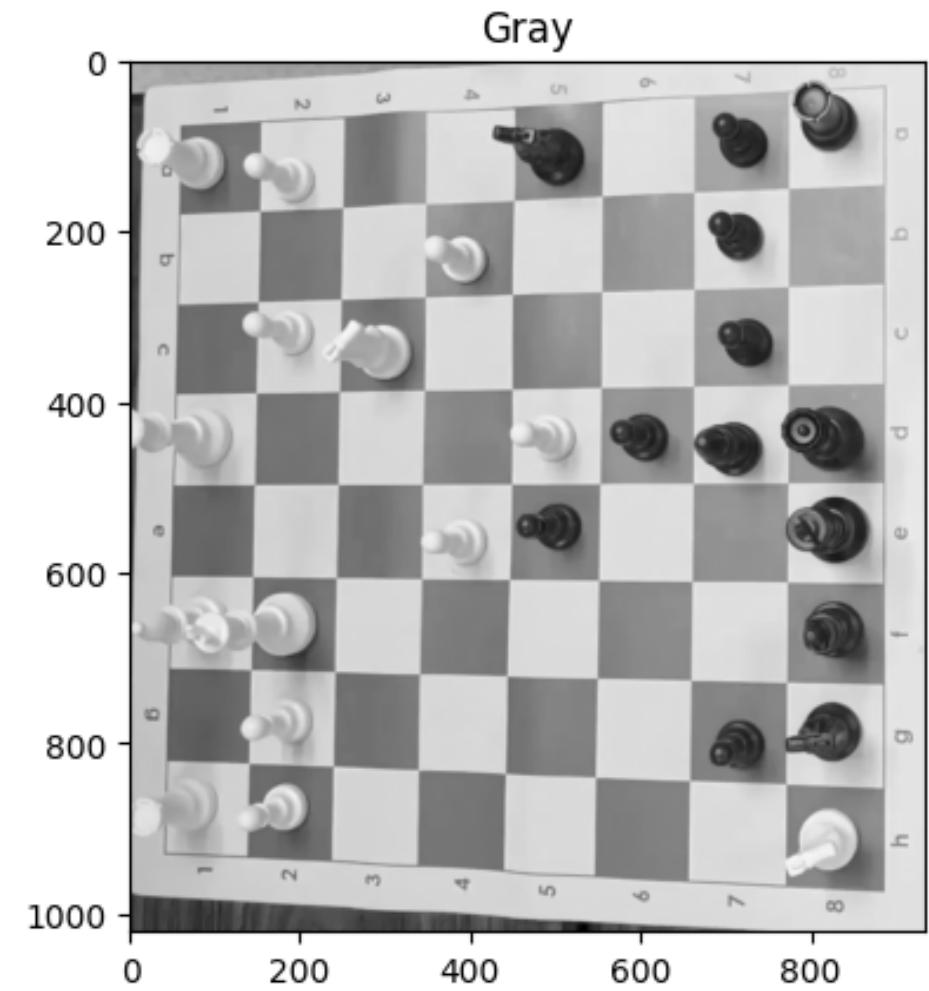
ใช้ Black Mask เพื่อตรวจจับ
ตำแหน่งของตัวหมากสีดำ



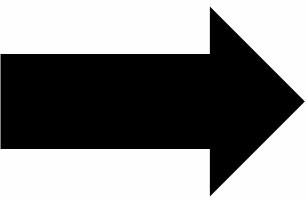
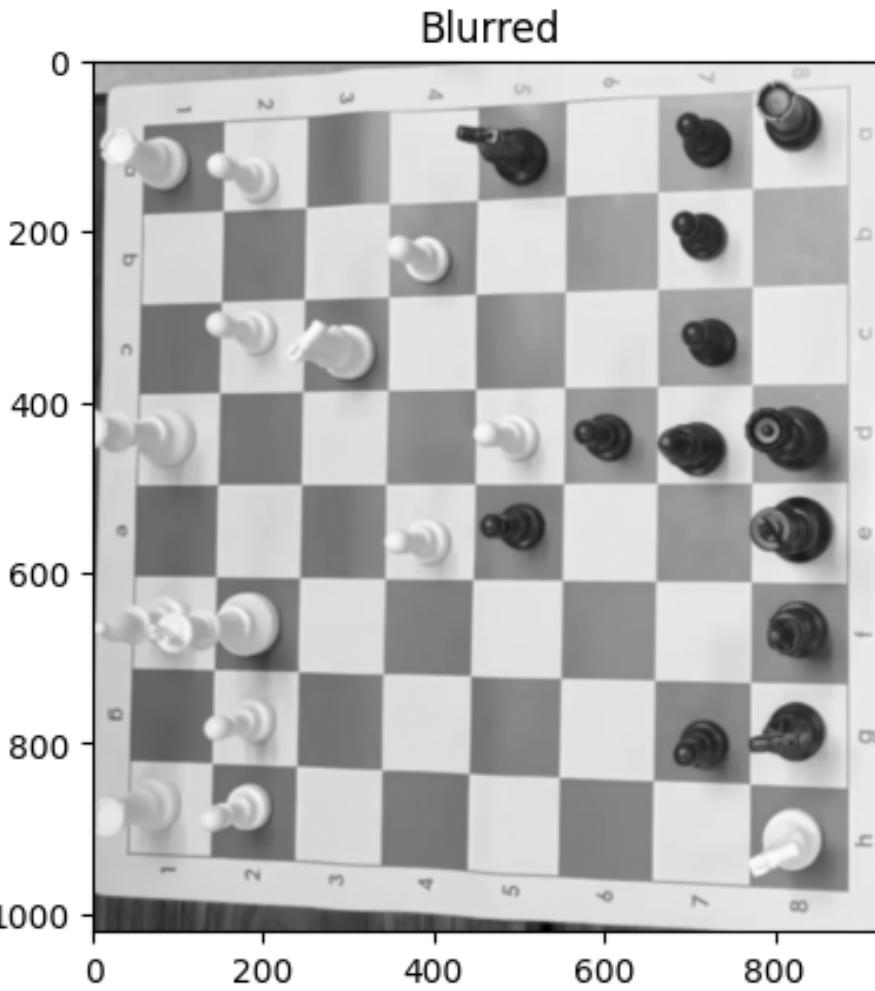


Chess Board & Piece Detection

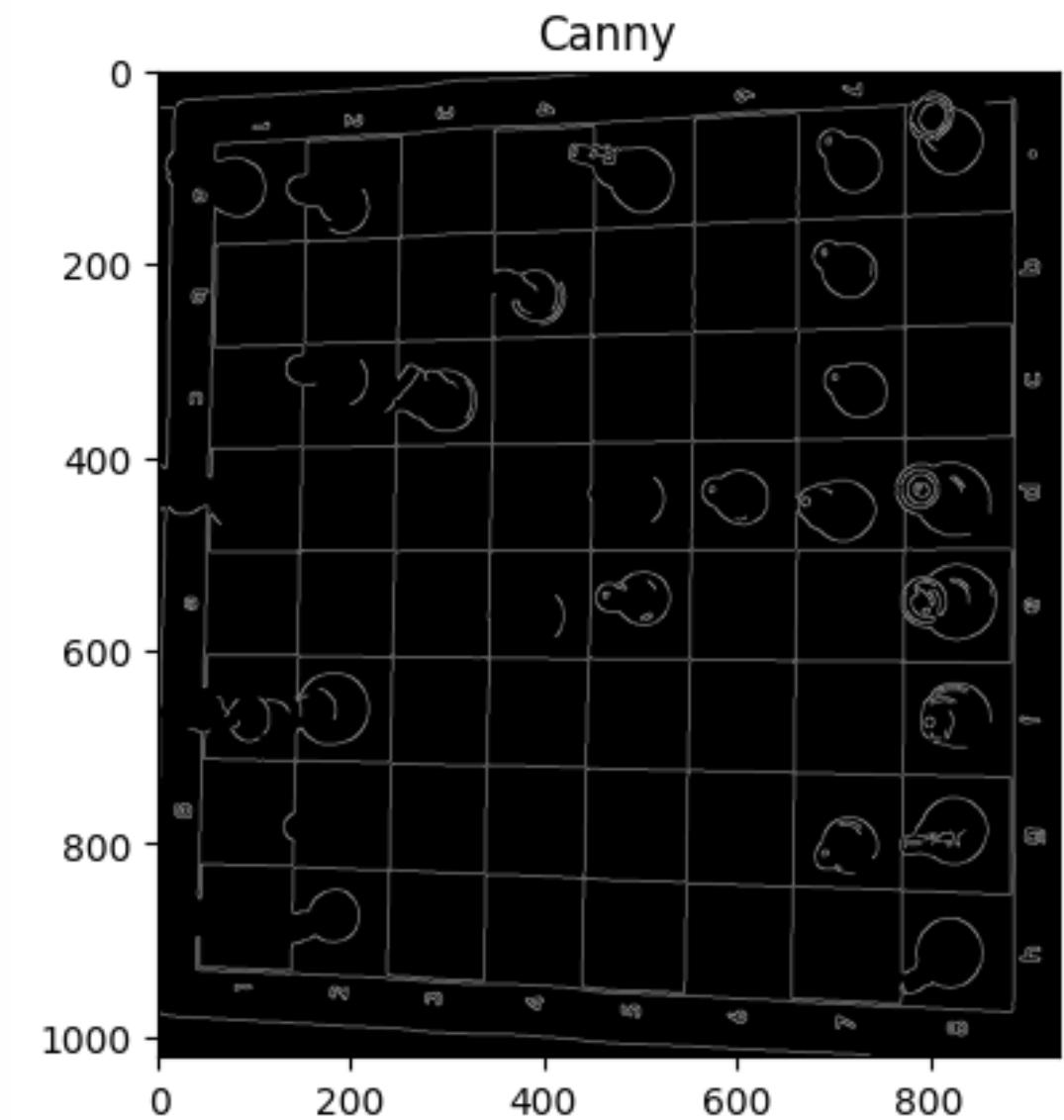
Grayscale & Blur



Blurred



Canny Edge Detection



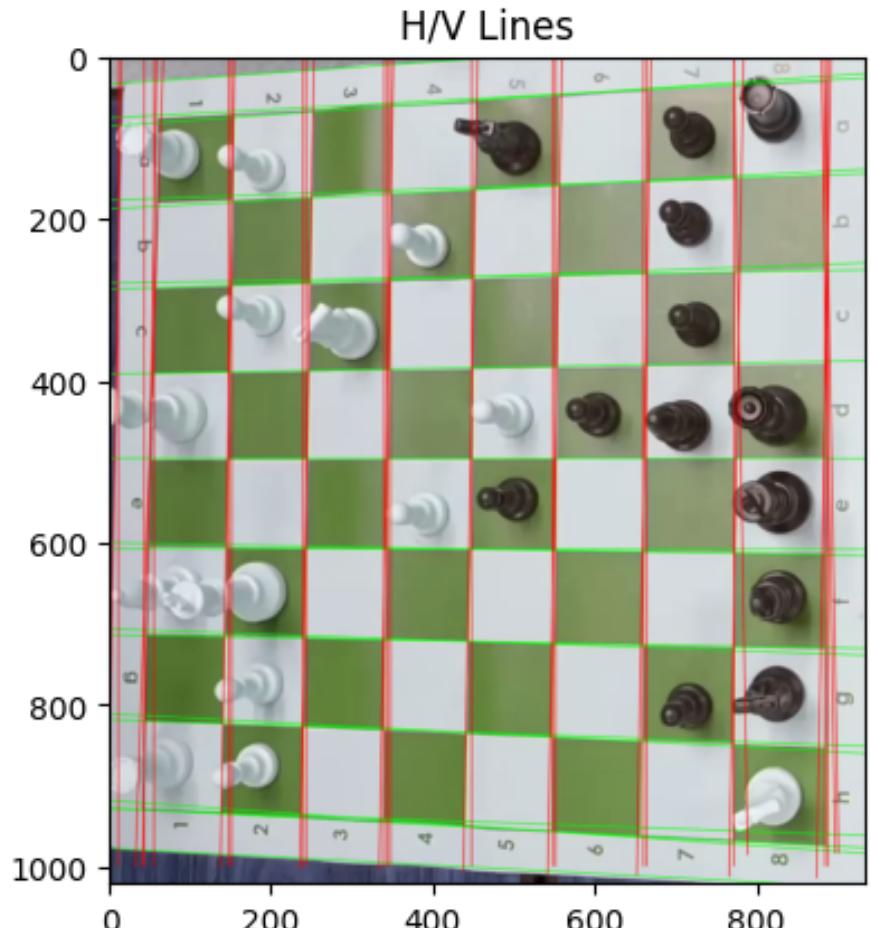
แปลงภาพเป็น Grayscale, ใช้ Gaussian Blur เพื่อลด Noise ให้โฟกัสแค่ "แสงและเงา"

ใช้ Canny Edge Detector ในการหาขอบที่ชัดเจน (ใช้ Otsu หาค่าที่เหมาะสมอัตโนมัติ)

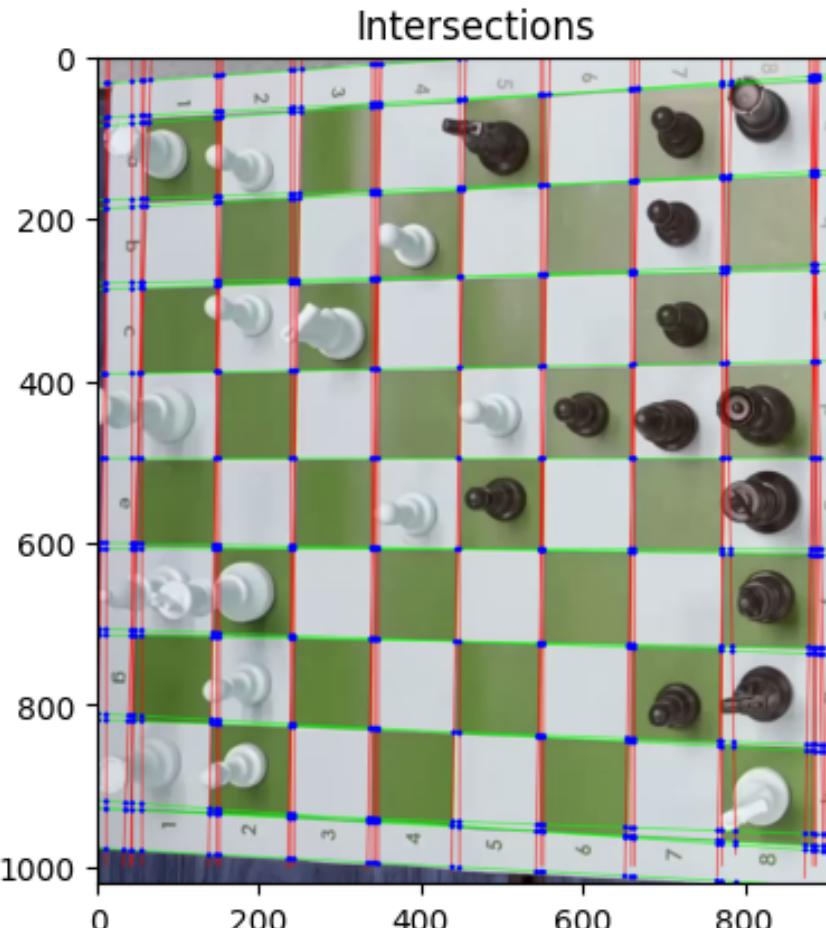


Chess Board & Piece Detection

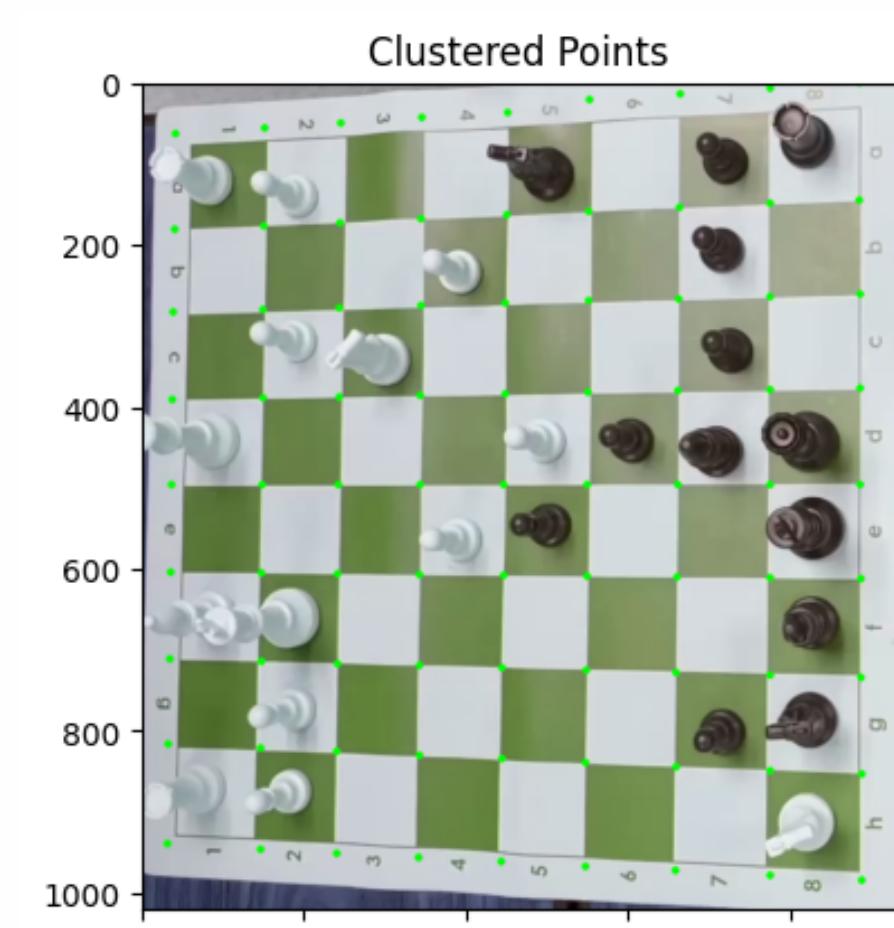
Hough Transform & Sort Lines



Calculate Intersections



Cluster Intersections



ใช้ Hough Transform
เพื่อตรวจจับเส้นตรงบนกระดาน
แยกเส้นที่เจือออกเป็น Horizontal (h)
และ Vertical (v) (ดูจากองศาของเส้น)

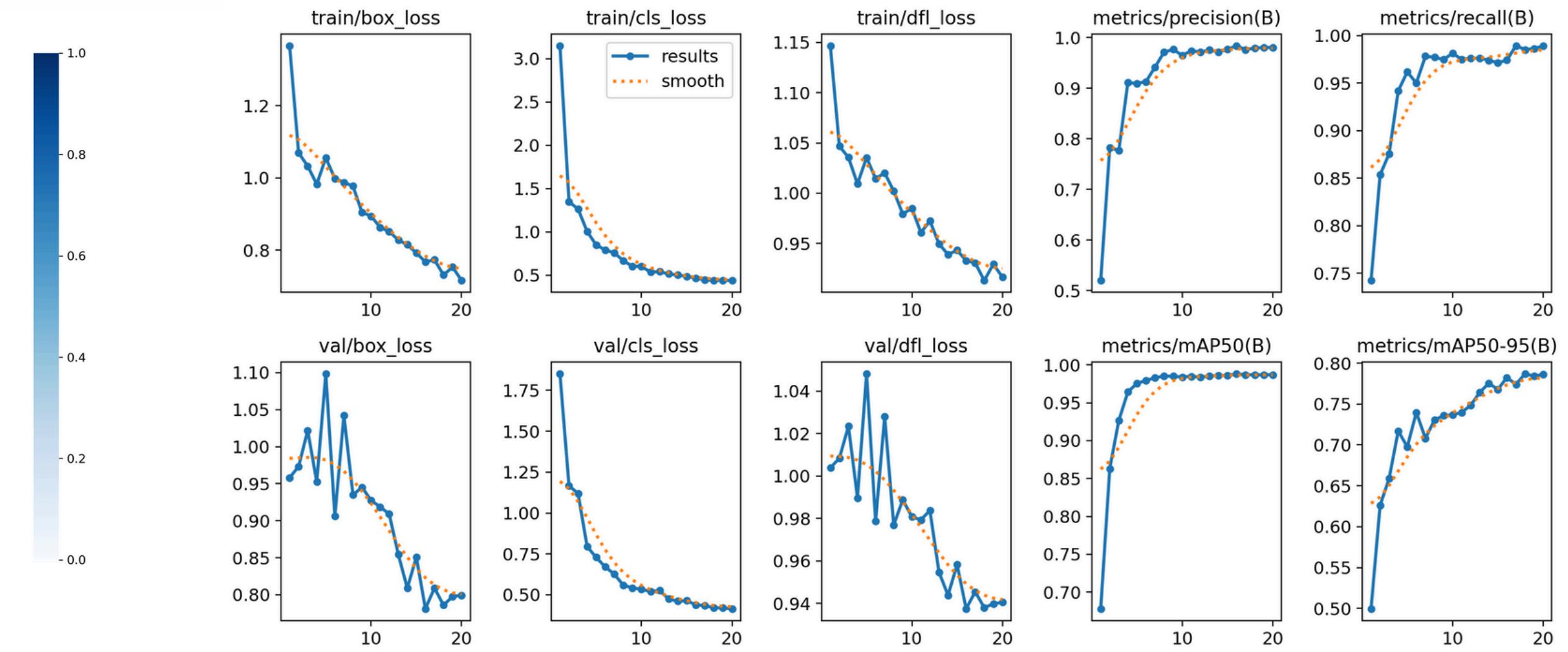
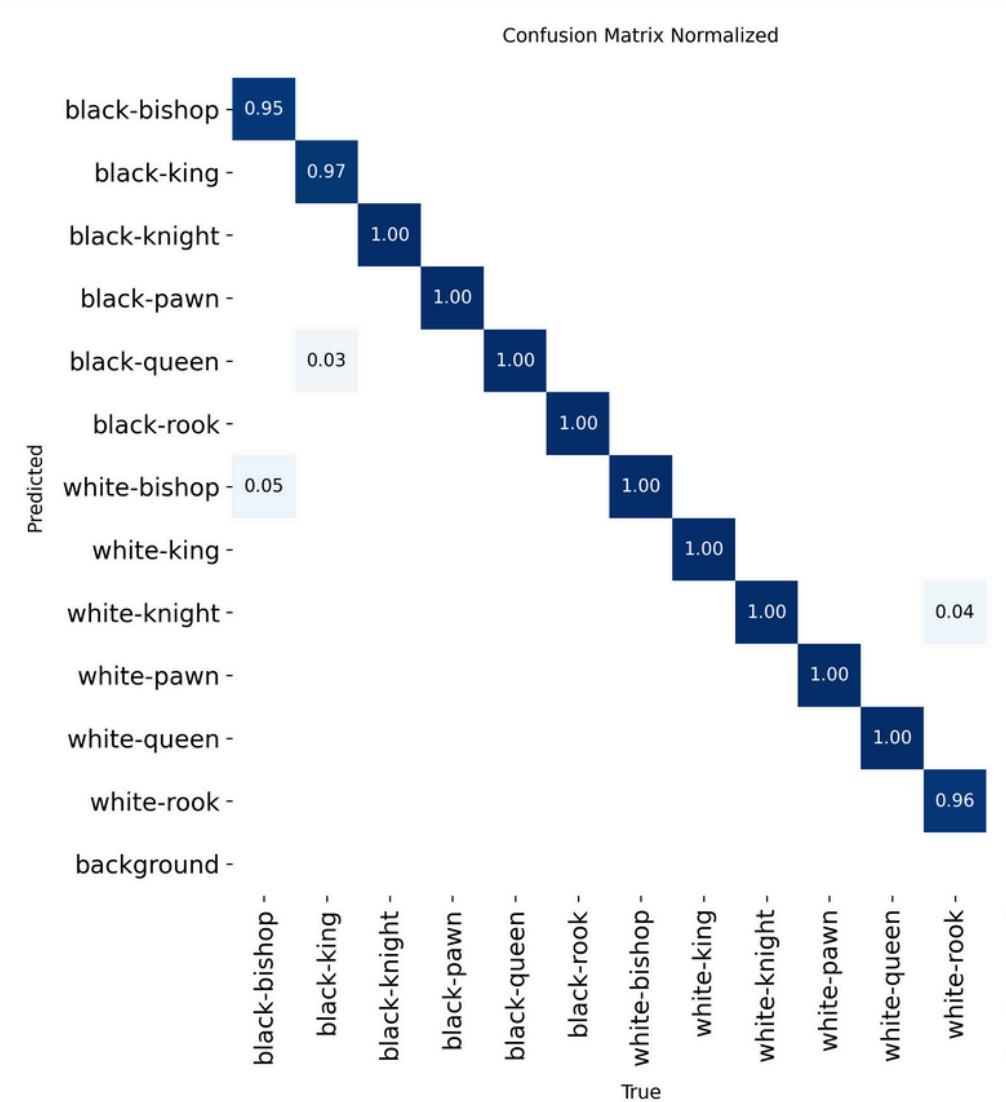
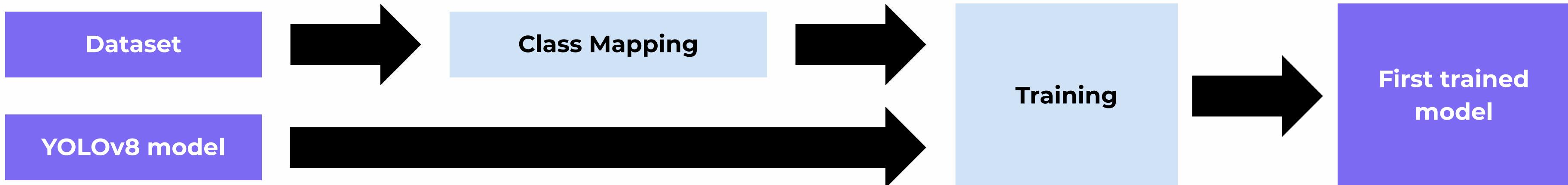
หาจุดตัดของเส้น h และ v →
ผลลัพธ์ได้หลายจุดเกินไป

Find Corners 4 มุมของกระดาน

จัดกลุ่มจุดตัดที่ใกล้เคียงกัน
เพื่อให้ได้จุดตัดที่ถูกต้อง

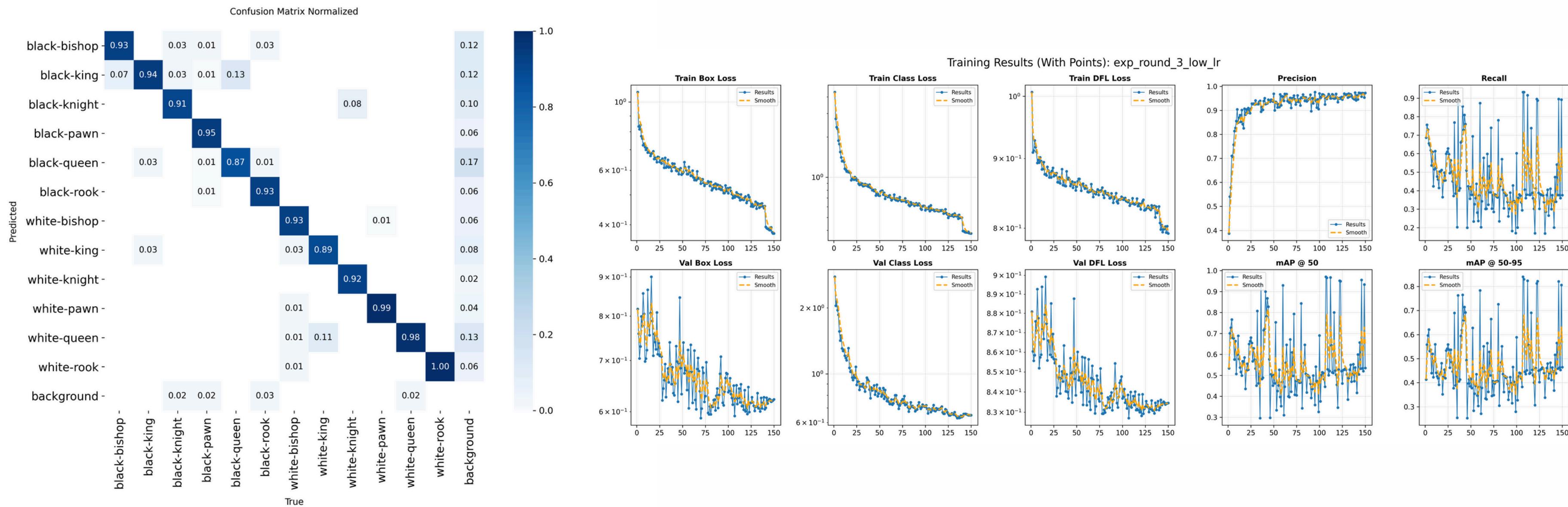
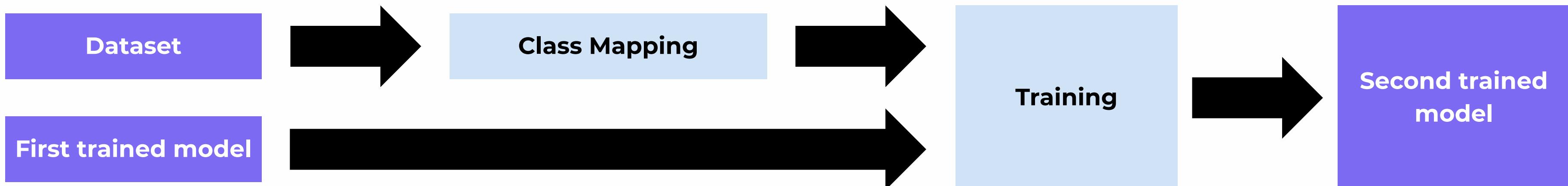


Chess Piece Detection



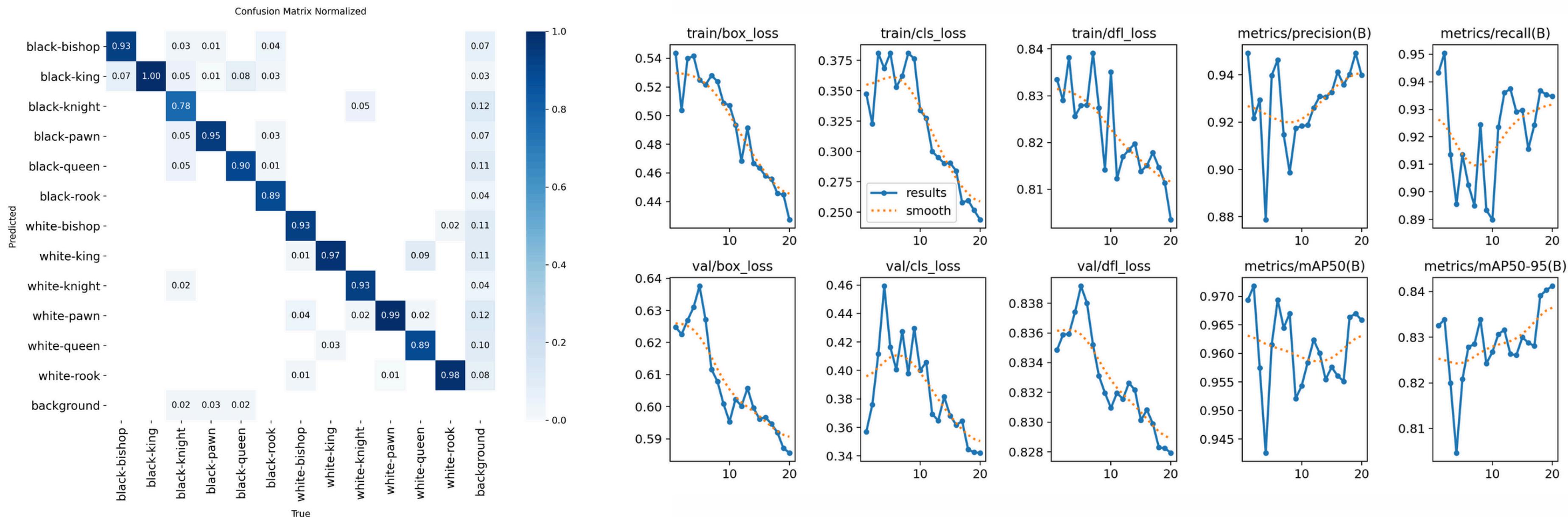


Chess Piece Detection



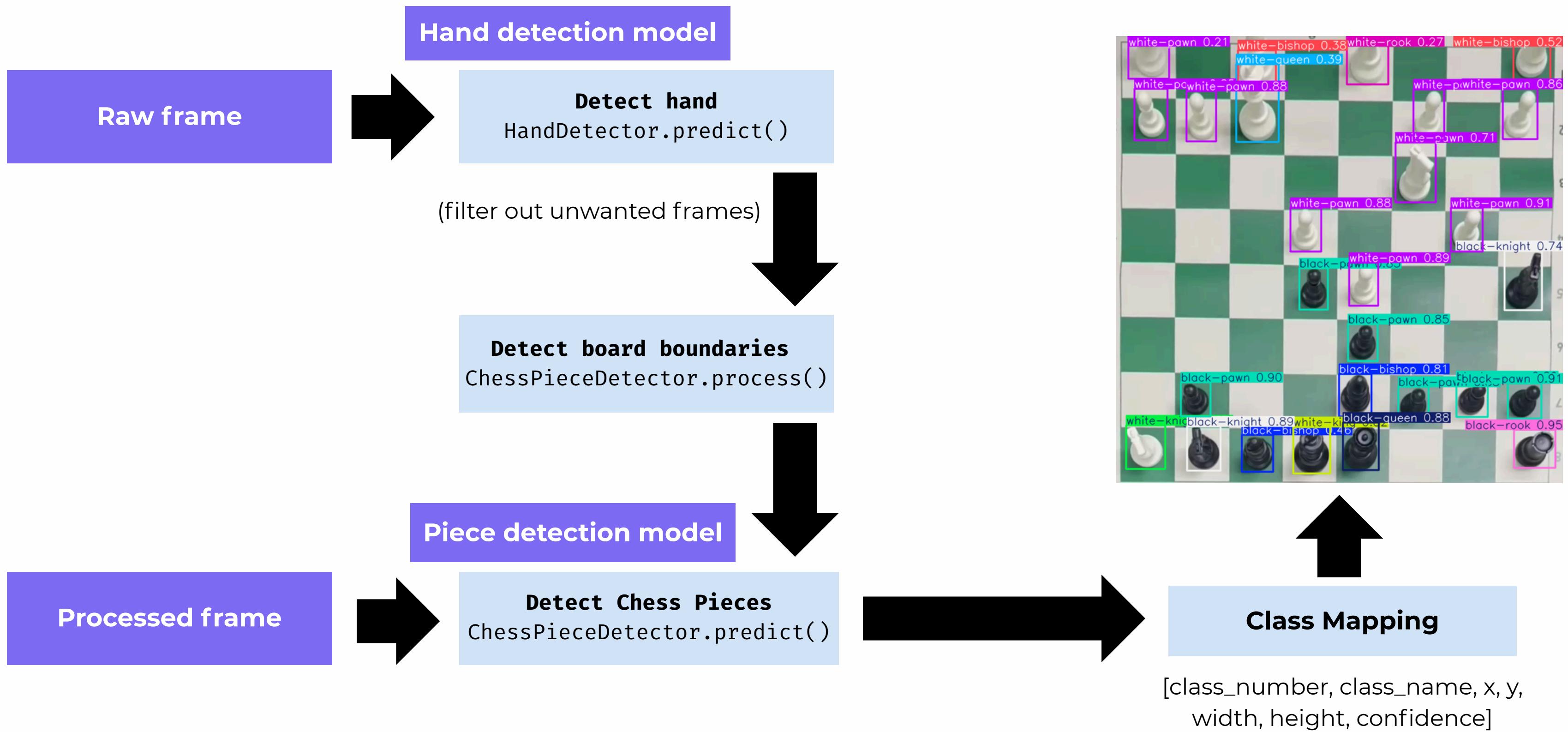


Chess Piece Detection





Chess Piece Detection





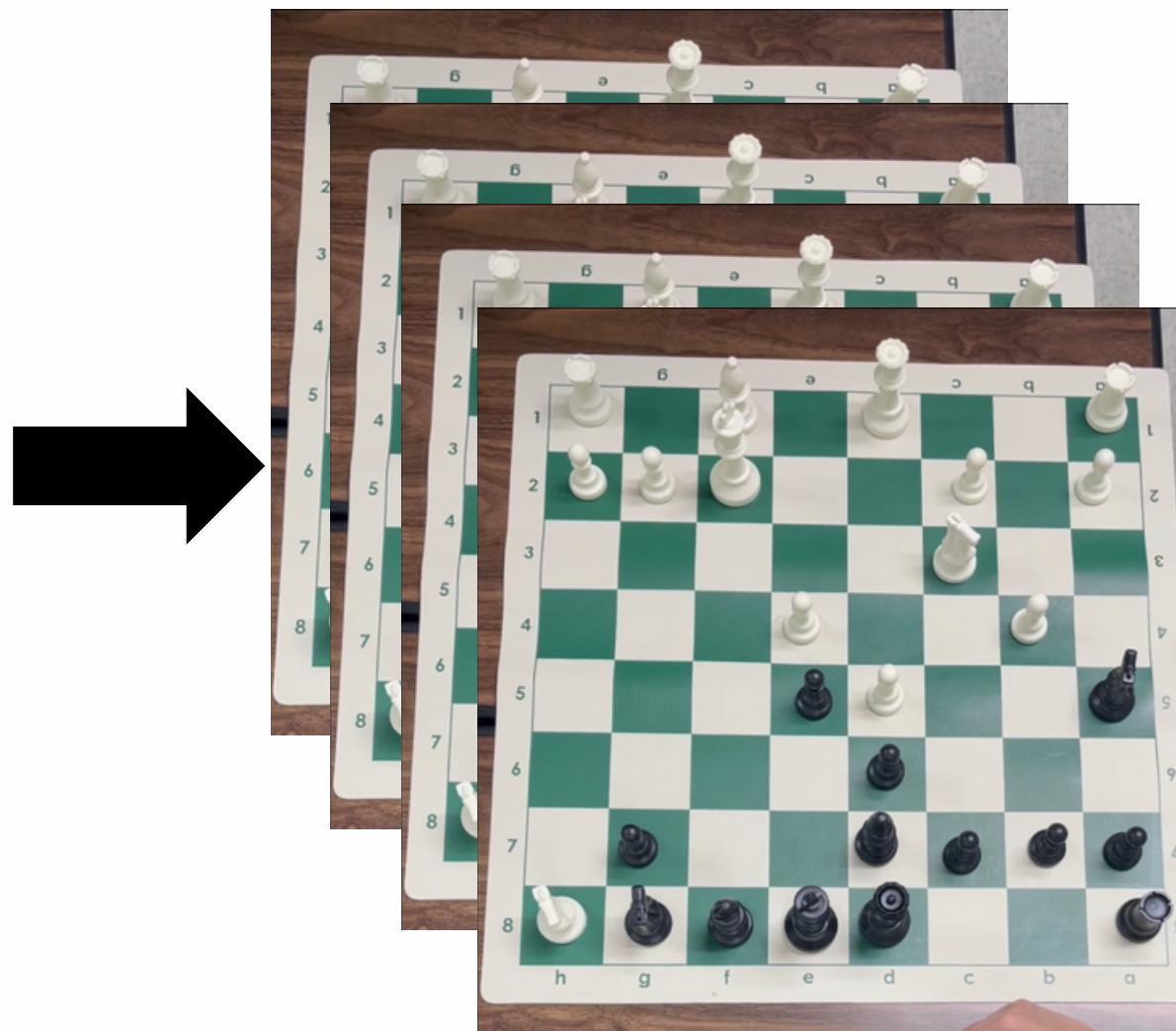
Chess Logic and Post-Processing

detection_to_chess_coordinates()

แปลงจากตำแหน่งพิกเซล เป็นพิกัด (ตำแหน่งช่อง) บนตารางหมากกระดาน



VIDEOS



STABLE FRAMES

```
Generating moves from 6 stable frames...
Chess coordinates of frame1: [[5, 7, 7], [3, 6, 7], [9, 3, 6], [4, 7, 4], [3, 6, 1],
Chess coordinates of frame2: [[5, 7, 7], [4, 7, 4], [9, 3, 6], [3, 6, 7], [3, 6, 1],
Chess coordinates of frame3: [[9, 3, 6], [3, 6, 1], [3, 6, 7], [3, 6, 6], [3, 5, 4],
Chess coordinates of frame4: [[5, 7, 7], [9, 3, 6], [3, 6, 1], [3, 6, 7], [3, 6, 6],
Chess coordinates of frame5: [[5, 7, 7], [8, 7, 0], [9, 3, 6], [2, 7, 1], [3, 6, 6],
Chess coordinates of frame6: [[5, 7, 7], [3, 6, 1], [8, 7, 0], [9, 3, 6], [2, 7, 1],
```

COORDINATES

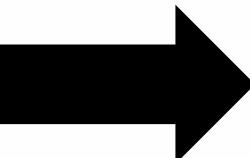


create_board()

สร้าง OBJECT ของ CHESS BOARD จากข้อมูลพิกัดของ แต่ละหน้า ด้วย LIBRARY PYTHON-CHESS

```
Generating moves from 6 stable frames...
Chess coordinates of frame1: [[5, 7, 7], [3, 6, 7], [9, 3, 6], [4, 7, 4], [3, 6, 1],
Chess coordinates of frame2: [[5, 7, 7], [4, 7, 4], [9, 3, 6], [3, 6, 7], [3, 6, 1],
Chess coordinates of frame3: [[9, 3, 6], [3, 6, 1], [3, 6, 7], [3, 6, 6], [3, 5, 4],
Chess coordinates of frame4: [[5, 7, 7], [9, 3, 6], [3, 6, 1], [3, 6, 7], [3, 6, 6],
Chess coordinates of frame5: [[5, 7, 7], [8, 7, 0], [9, 3, 6], [2, 7, 1], [3, 6, 6],
Chess coordinates of frame6: [[5, 7, 7], [3, 6, 1], [8, 7, 0], [9, 3, 6], [2, 7, 1],
```

COORDINATES



```
Frame 1 board:
r . . .
. . . .
. . . .
. . . .
. . . .
. . . .

Frame 2 board:
. . . .
. . . .
. . . .
. . . .
. . . .
. . . .

Frame 3 board:
. . . .
. . . .
. . . .
. . . .
. . . .
. . . .

Frame 4 board:
. . . .
. . . .
. . . .
. . . .
. . . .
. . . .

Frame 5 board:
. . . .
. . . .
. . . .
. . . .
. . . .
. . . .

Frame 6 board:
. . . .
. . . .
. . . .
. . . .
. . . .
. . . .

Pawn: P
Rook: R
Knight: N
Bishop: B
Queen: Q
King: K
```

BOARDS



Chess Logic and Post-Processing

boards_to_pgn_move()

DETECT และ VALIDATE MOVE

- เทียบสถานะก่อนหน้าและปัจจุบันของ BOARD
- ใช้ช่องที่มากหายไปและช่องใหม่ที่มากปราภูมาร้างเป็น MOVE
- มีการ VALIDATE กับเกติกาหมายกว่าเดินถูกต้องหรือไม่
- ถ้าถูกต้องก็จะแปลง MOVE นั้นเป็น PGN FORMAT

Frame 1 board:
r Frame 2 board:
p |r Frame 3 board:
· P r Frame 4 board:
n · P r Frame 5 board:
· |n · P r Frame 6 board:
· · n · P r . . . K B n N
P · · [n · p p p B . . p .
B P . . n . . p . . .
R P . . n . . P p . . .
B P . . P . . P . . q
B P . . P . . P . .
B P . P . . Q . P
B . . Q . B . .



... bxa2 gxa2

PGN FORMAT

BOARDS



PGN Generation & Output

การสร้างรายการการเดินหมาก PGN และการส่งออกผลลัพธ์

▶ **ฟังก์ชันหลัก:** frames_to_pgn_movelist() และ export_csv_result()

ขั้นตอนการสร้าง PGN:

- วนซ้ำผ่านเฟรม (Frames) ของวิดีโอทั้งหมด
- ติดตามสถานะกระดาน (last_valid_board) และจำนวนตาเดิน (game_turn)
- ใช้ผลลัพธ์จาก boards_to_pgn_move() มาต่อกันเป็น PGN Movelist (เช่น 1. e4 e5 2. Nf3 Nc6...)
โดยคำนึงถึงตาเดินของขาว/ดำ

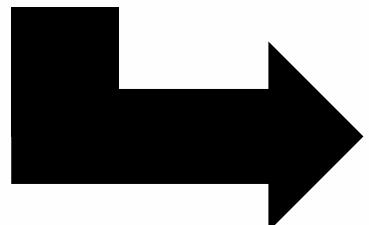
การส่งออก (Output):

export_csv_result(): จัดเก็บผลลัพธ์ ([clip_name, pgn_movelist]) ลงในไฟล์ CSV เพื่อส่งงาน



Evaluation

เชิงปริมาณ (QUANTITATIVE)

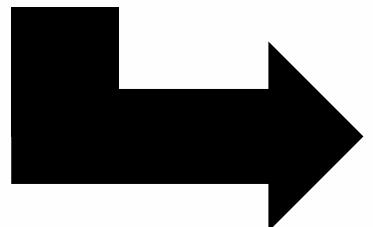


- **Metric :** Edit Distance
- การวัดความแตกต่างระหว่างการเดินหมากที่ระบบทำนาย (Your Result) กับการเดินหมากที่ถูกต้อง (Solution) ค่าต่ำหมายถึงความแม่นยำสูง (0 คือถูกต้อง 100%)
- **ตัวอย่างผลลัพธ์จาก Project**
 - Your Result: 1. g3
 - Solution: 1. Qh4+ 2. g3
 - Edit Distance: 0.384



Evaluation

เชิงคุณภาพ (QUALITATIVE)



- **Metric :** Confidence Score / Visual Inspection
- ความเชื่อมั่นของโมเดลต่อการตรวจจับตัวหมาย และการตรวจสอบด้วยตาว่าระบบสามารถปรับมุมมองและระบุตำแหน่งหมายได้อย่างถูกต้องหรือไม่
- **ตัวอย่างผลลัพธ์จาก Project**
 - Confidence Score สูงสุดถึง 0.94 สำหรับตัวหมายที่ตรวจพบ



Evaluation

Pro & Con of Selected Technique

Hough Transform & Clustering (สำหรับตรวจจับกระดาน)

- แม่นยำสูง ในการหาเส้นตรงแม้มีสัญญาณรบกวน (NOISE)
- ทำงานได้ รวดเร็ว บน CPU เมื่อเทียบกับ DL-BASED BOARD DETECTION
- อ่อนไหวต่อเงา/แสง ที่ทำให้เกิด "ขอบ" ปลอมบนกระดาน
- ต้องการ PARAMETER ที่ต้องจูนอย่างละเอียด (เช่น THRESHOLD ของ HOUGH LINES)

YOLO/ULTRALYTICS (สำหรับตรวจจับตัวมาก)

- ความเร็วสูง (REAL-TIME) เหมาะสำหรับการวิเคราะห์วิดีโอ
- แม่นยำสูง ในการตรวจจับและจำแนกวัตถุ (ตัวมาก) หลายชั้นพร้อมกัน
- ต้องใช้ DATASET ขนาดใหญ่ สำหรับการฝึกฝนมโนเดล
- ต้องการ GPU ใน การประมวลผล (TRAINING/INFERENCE) เพื่อให้ได้ความเร็วสูงสุด



Model Evaluation

mAP50: 0.9658
mAP50-95: 0.8416
Precision: 0.9366
Recall: 0.9406

Dataset ที่ใช้ Train Model คือ **Roboflow --> Dataset อาจารย์: chess_label_dataset --> Dataset อาจารย์ (กระดานเปล่า): boardchess**

- **ความแม่นยำโดยรวม (mAP):** ดีเยี่ยม
- **ความแม่นยำของกรอบตัวแทน (mAP50-95):** ดีมาก (ไม่เดลสามารถตรวจ Bounding Box ได้อย่างแม่นยำ)
- **ความสมดุลระหว่าง Precision และ Recall:** สมดุลมาก (กึ่งสองค่าใกล้เคียง 0.94)

Kaggle Public Score

Submission and Description

Public Score ⓘ



Lovecat888_Code - 0.79_Omsin_Fixed_ChessLogic

Complete · Chanatda Konchom · 7h ago

0.80



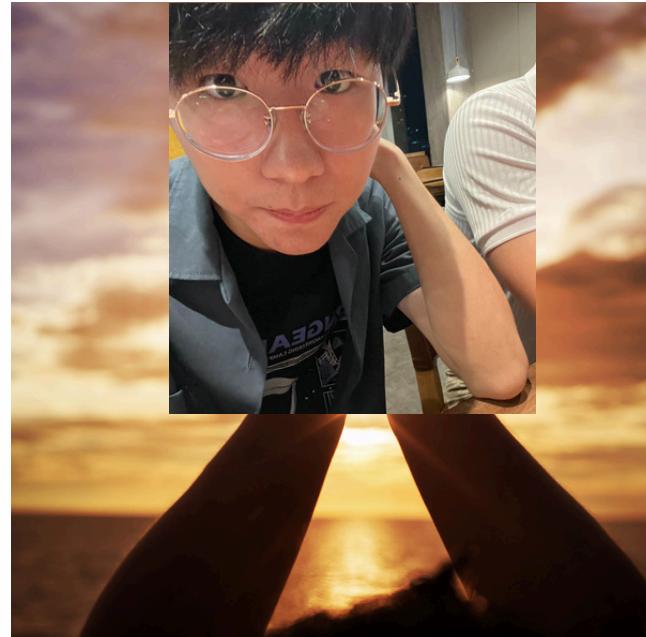
Roles & Responsibilities



Tee

PIYAPHON EKAPHOPNARONG

- TRAIN CHESS PIECE MODEL
- CREATE THE RESULT
- SLIDE PRESENTATION



Faith

CHAYUT ARCHAMONGKOL

- TRAIN CHESS PIECE MODEL
- SLIDE PRESENTATION



Omsin

KRISSADA SINGHAKACHAIN

- TRAIN CHESS PIECE MODEL
- SLIDE PRESENTATION



Fai

CHANATDA KONCHOM

- CODING PIPELINE
- RESEARCH DATASET
- RESEARCH PRETRAINED MODEL
- TRAIN CHESS PIECE MODEL

#NoGPUNoLife #digimageໄຟສາແກRTX5080

OutOfMemoryError: CUDA out of memory. Tried to allocate 20.00 MiB.

RuntimeError: MPS backend out of memory (MPS allocated: 17.64 GiB, other allocations: 502.92 MiB, max allowed: 18.13 GiB). Tried to allocate 256 bytes on shared pool.

OutOfMemoryError: CUDA out of memory. Tried to allocate 32.00 MiB. GPU 0 has a total capacity of 7.96 GiB of which 0 bytes is free.

#MostEntriesStillNewbies