



FINAL PROJECT

CHESS MOVE TRACKING

INTRODUCTION TO DIGITAL IMAGING (2110431)

TIMER: 31:14
MOVE: 13. hxg4



!!! FIDE WORLD CHAMPIONSHIP SINGAPORE 2024

GAME 3 / 14
FIRST TO 7.5 WINS

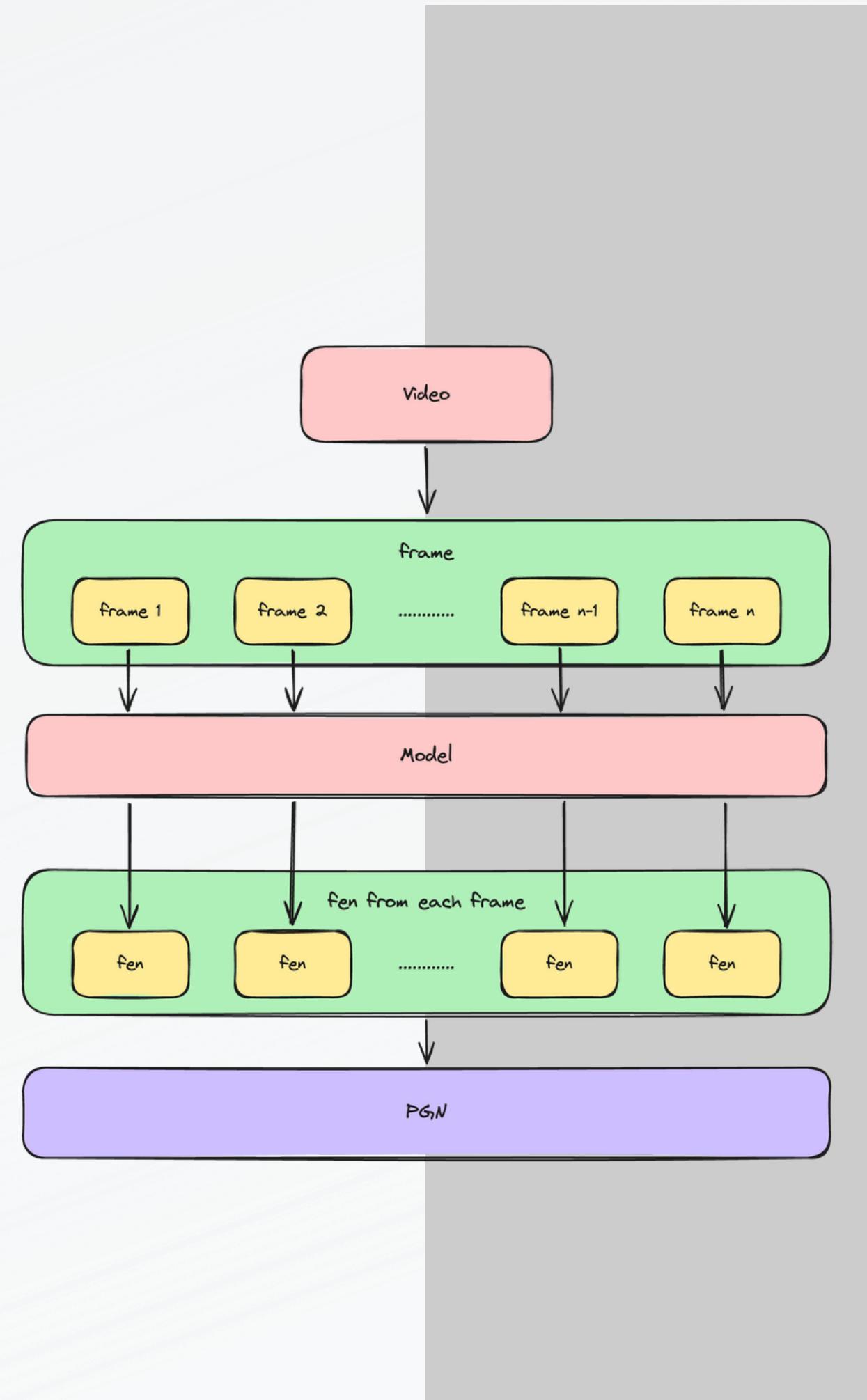


OUR APPROACH

1. Extract each frame from video

2. Input each frame to model

3. Convert FEN to PGN



MODEL

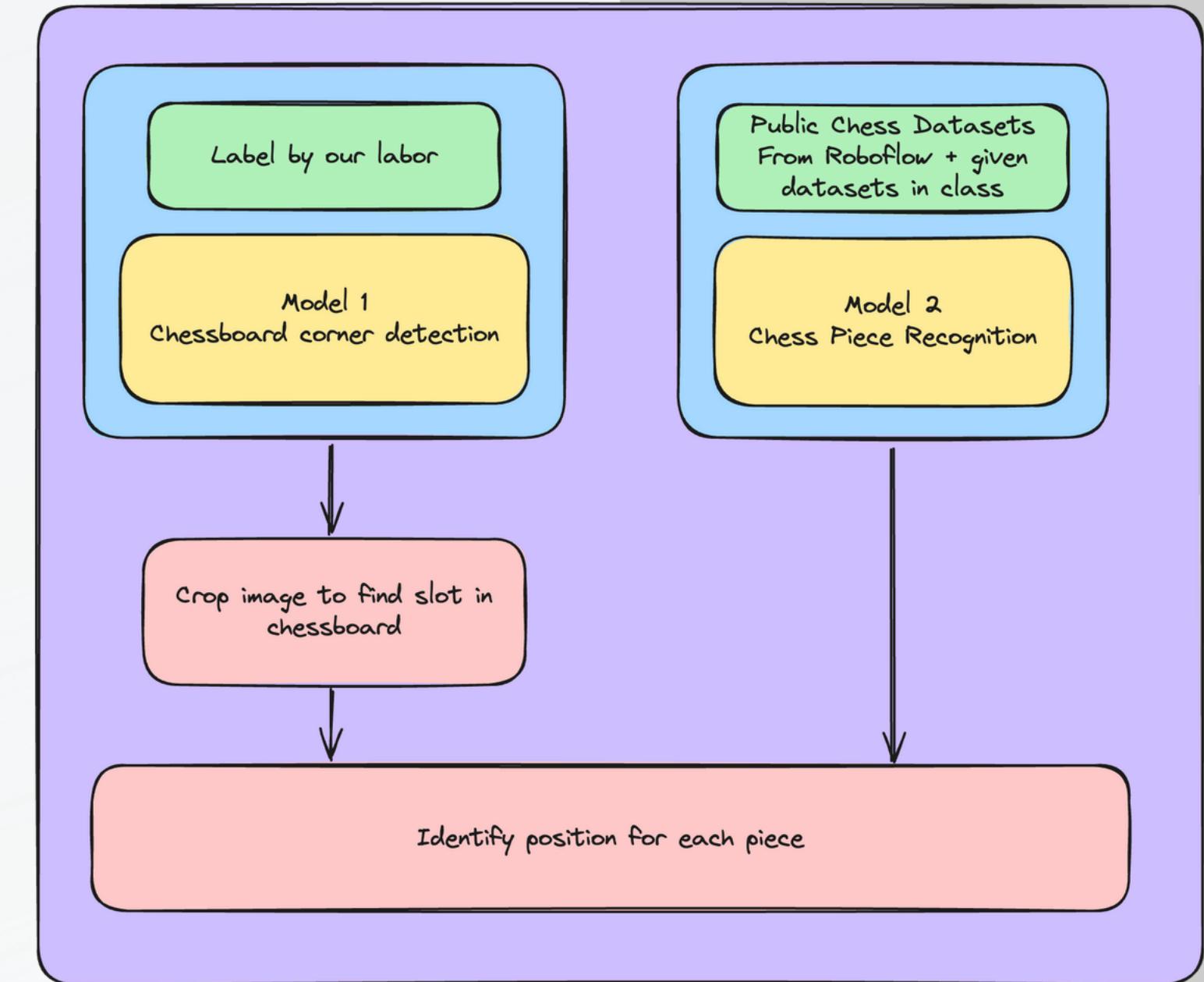
Model

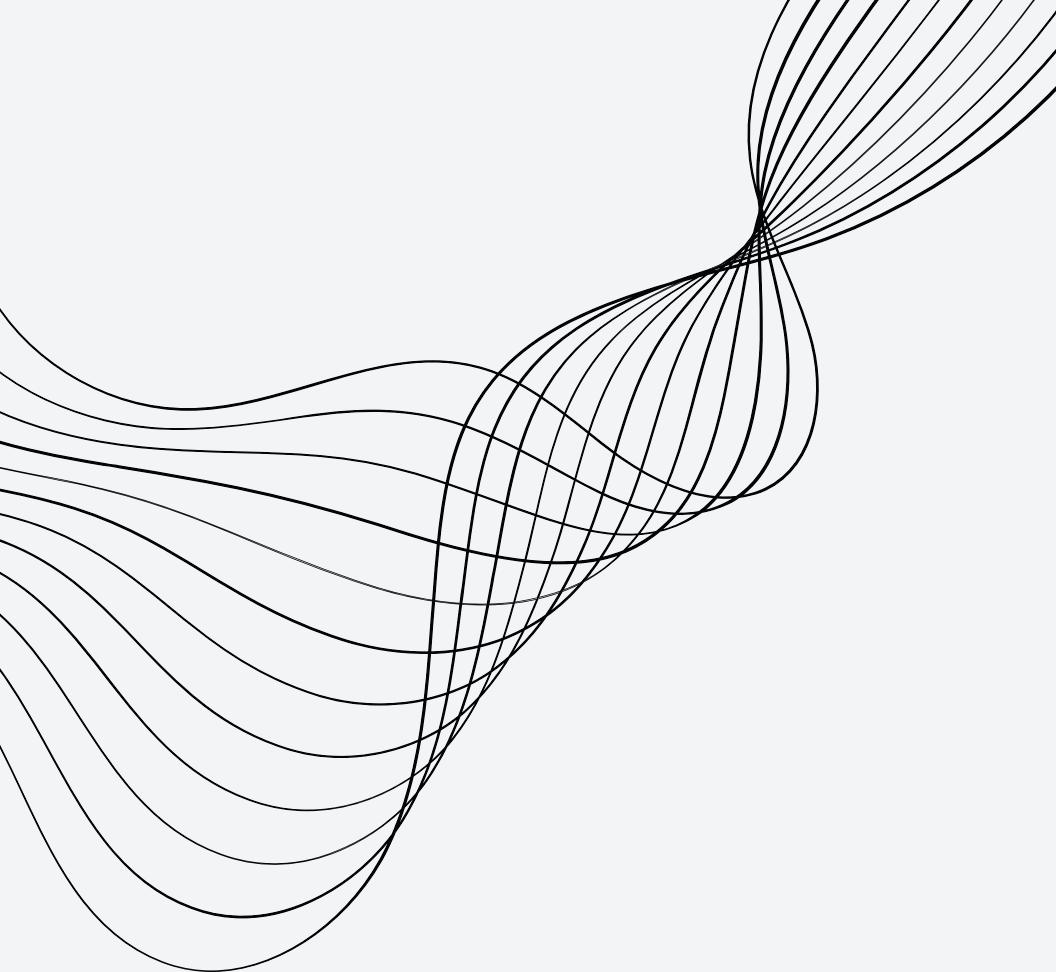
Chessboard corner
detection

then, crop image to find slot in chessboard and
identify position for each piece

Model

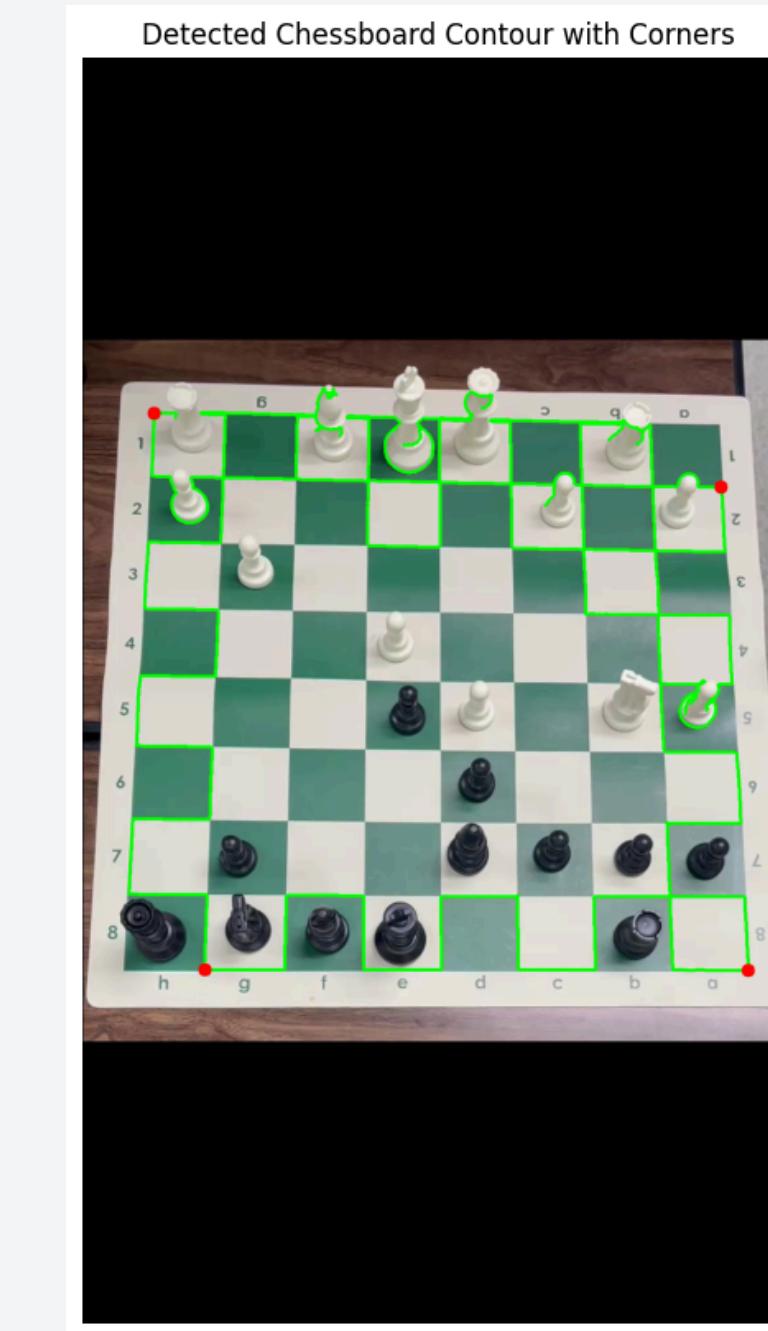
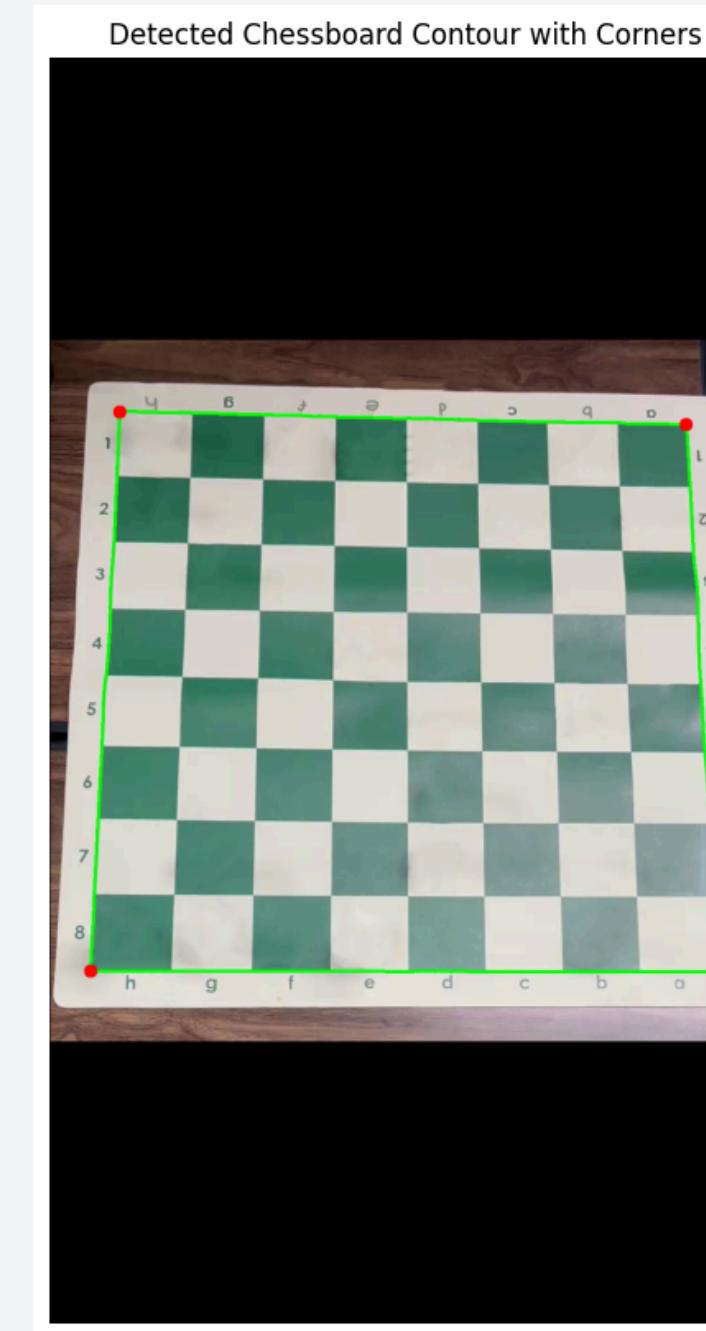
Chess Piece
Recognition





CHESSBOARD CORNER DETECTION

TRADITIONAL APPROACH

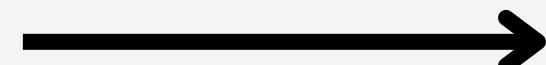


TRAINING MODEL

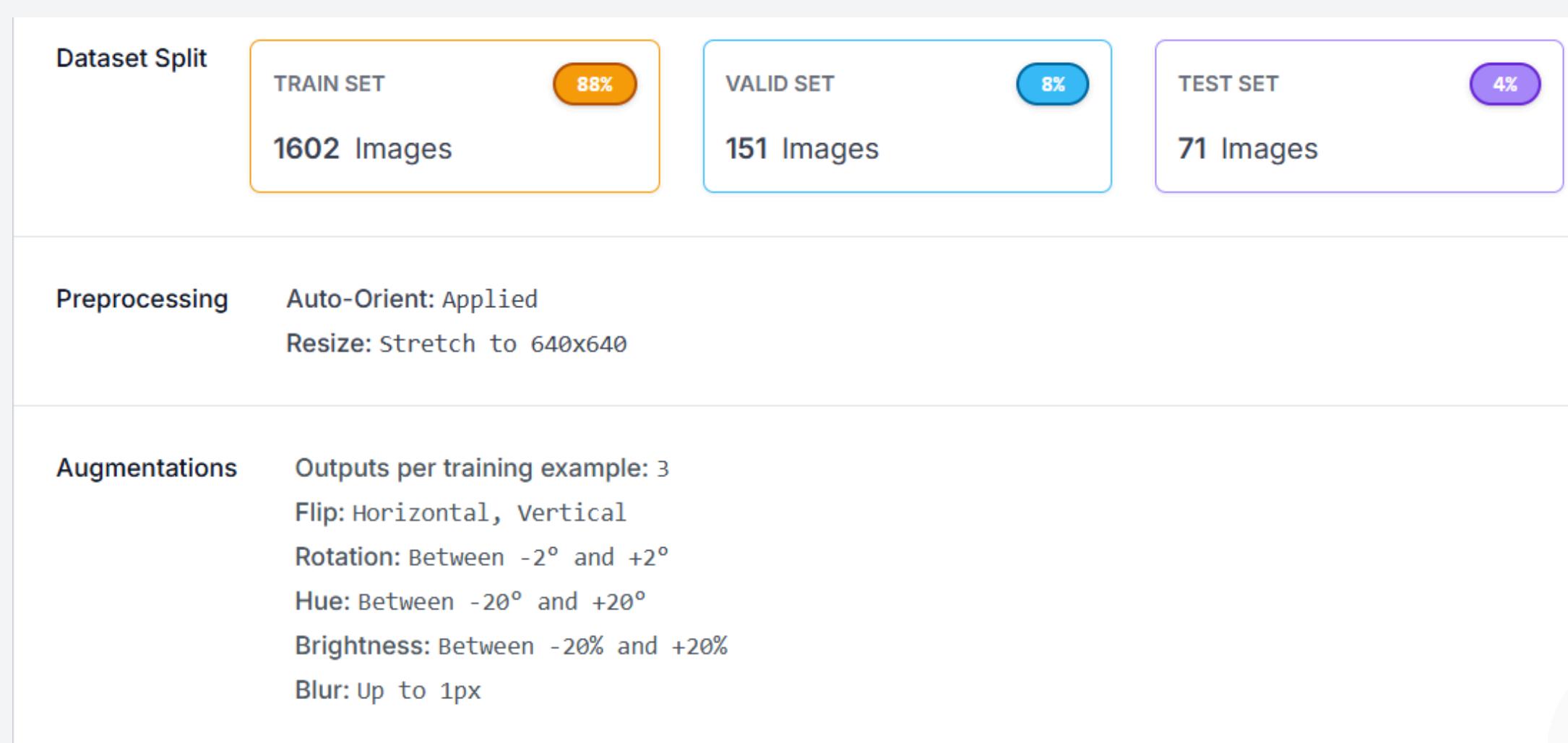
YOLOv8

100 epochs

32 batch size

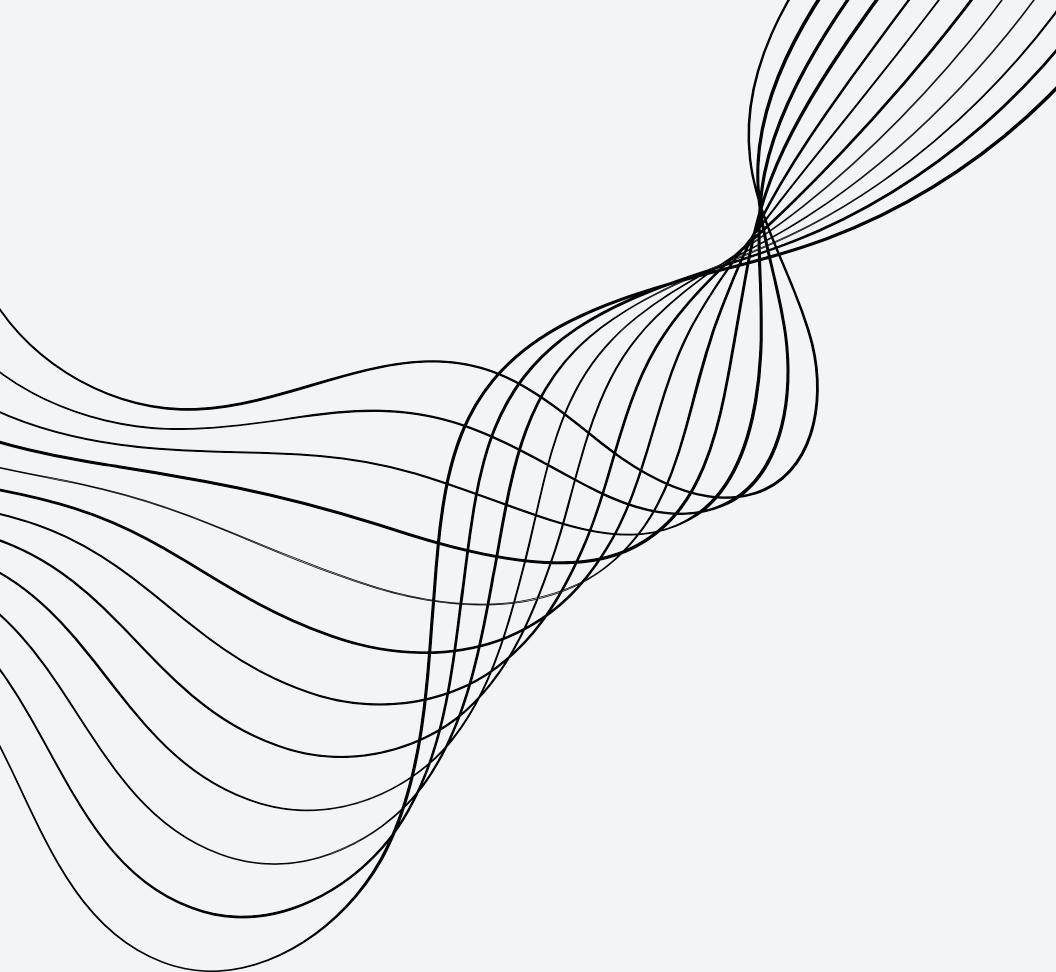


CORNER DATA



Dataset	14 Jobs	See all 756 images	⋮
Uploaded on 12/07/24 at 8:49 pm: Job 3	mazu roze	185 Images	🔗 📂
Uploaded on 12/07/24 at 8:49 pm: Job 2	Phumsiri Sumativit	186 Images	🔗 📂
Folder: images: Job 4	Phumsiri Sumativit	47 Images	🔗 📂
Folder: images: Job 3	Phumsiri Sumativit	40 Images	🔗 📂

MANUAL LABEL W. SUNNYSLEEPY



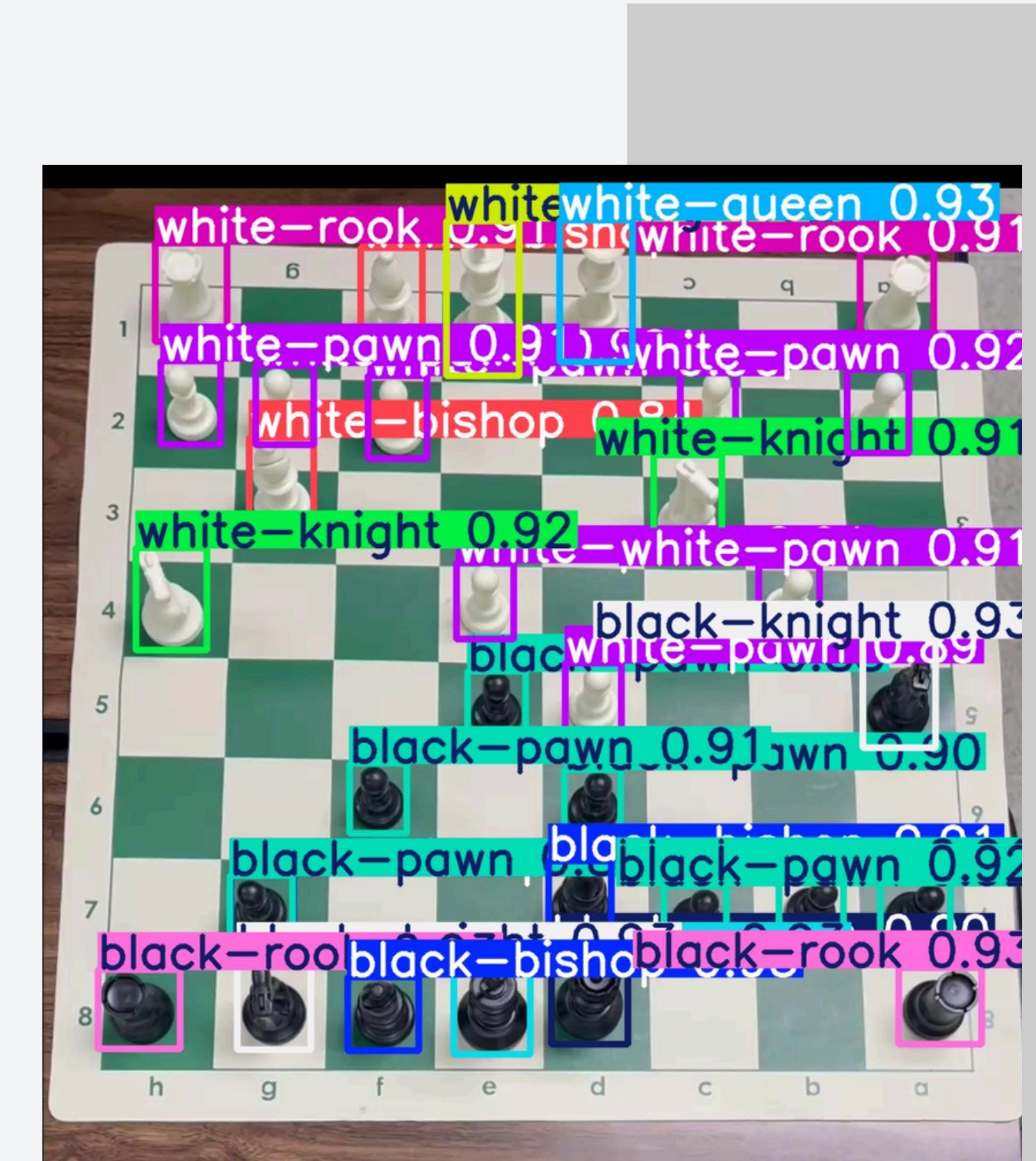
CHESS PIECE RECOGNITION

TRAINING MODEI

YOLOv11m

100 epochs

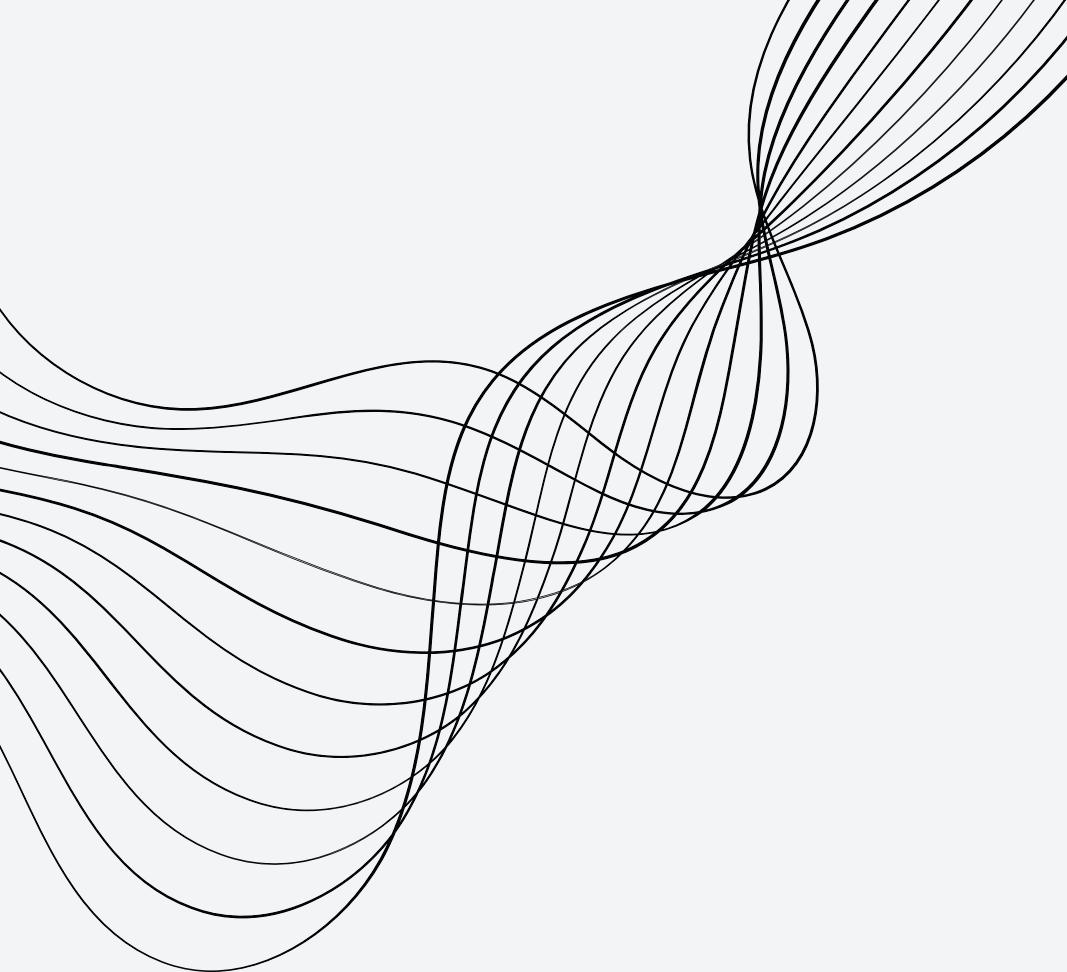
16 batch size



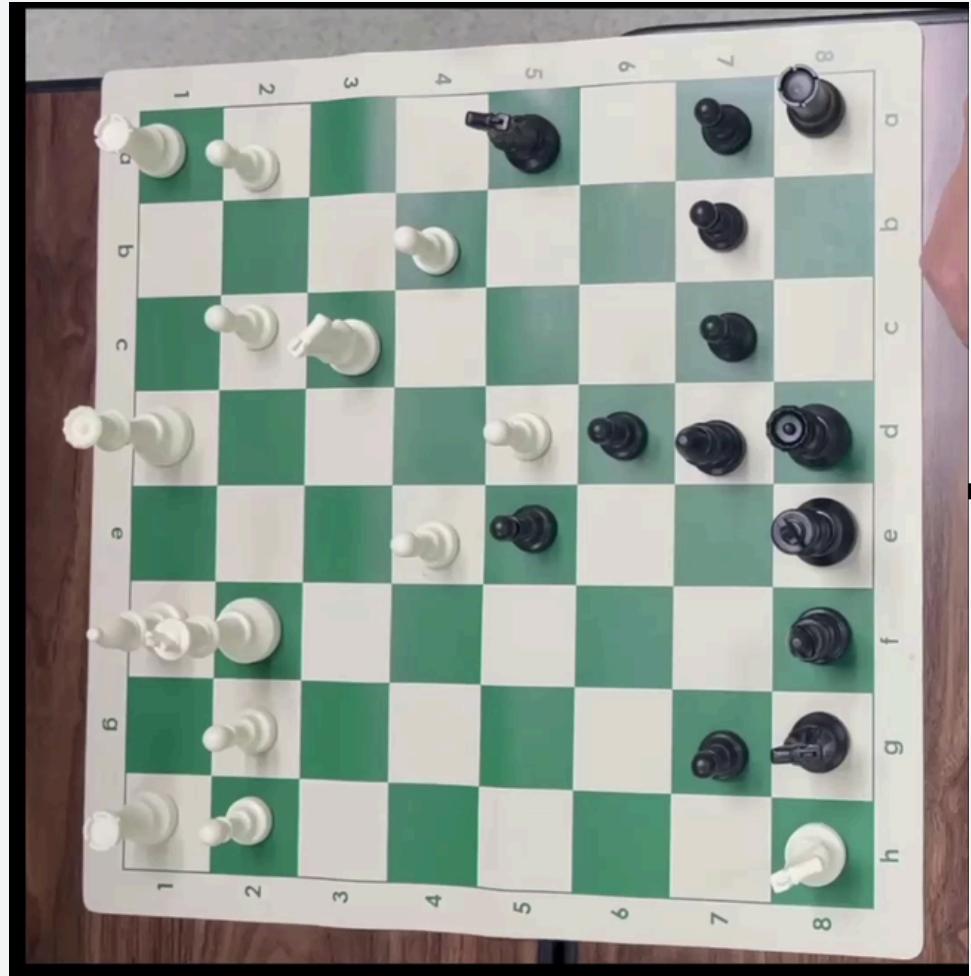
CHESS PIECE DATA

Dataset Split	TRAIN SET	88%	VALID SET	8%	TEST SET	4%
	2304 Images		216 Images		111 Images	
Preprocessing	Auto-Orient: Applied Resize: Stretch to 640x640					
Augmentations	Outputs per training example: 3 Flip: Horizontal, Vertical Hue: Between -20° and +20° Brightness: Between -25% and +25% Exposure: Between -10% and +10% Blur: Up to 0.5px Noise: Up to 0.3% of pixels					

ROBOFLOW + PROVIDED DATASETS



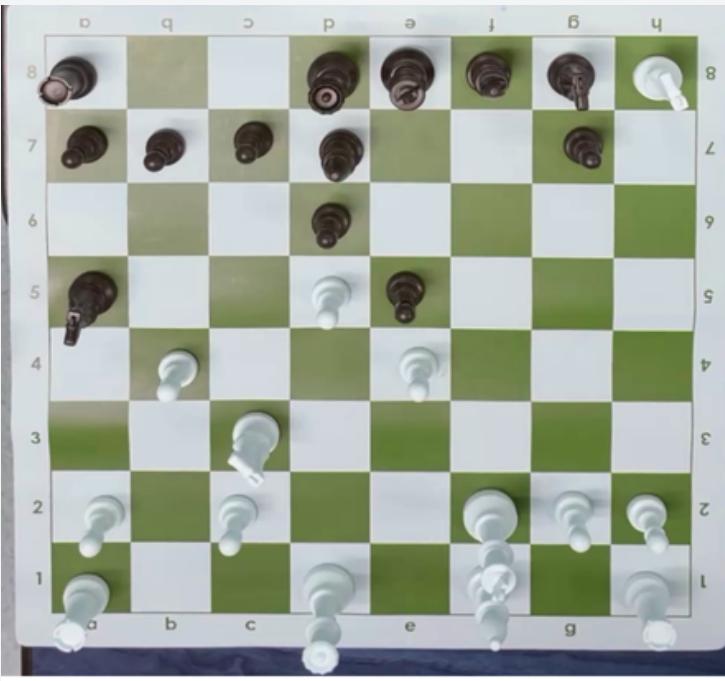
MAPPING TO DIGITAL BOARD



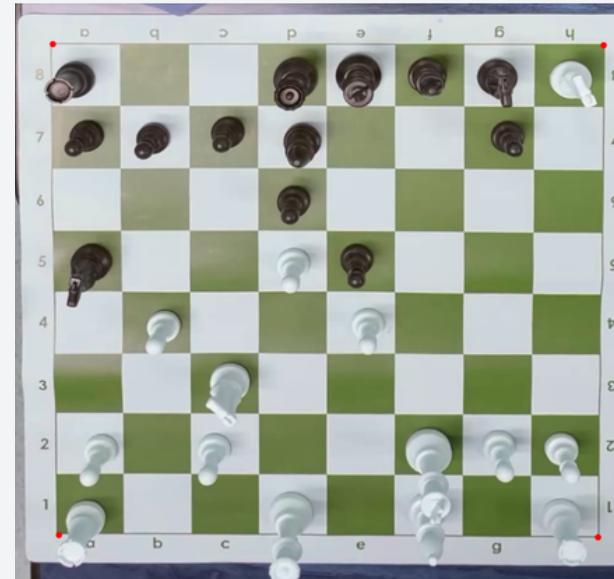
Transform

Chess Detection Model

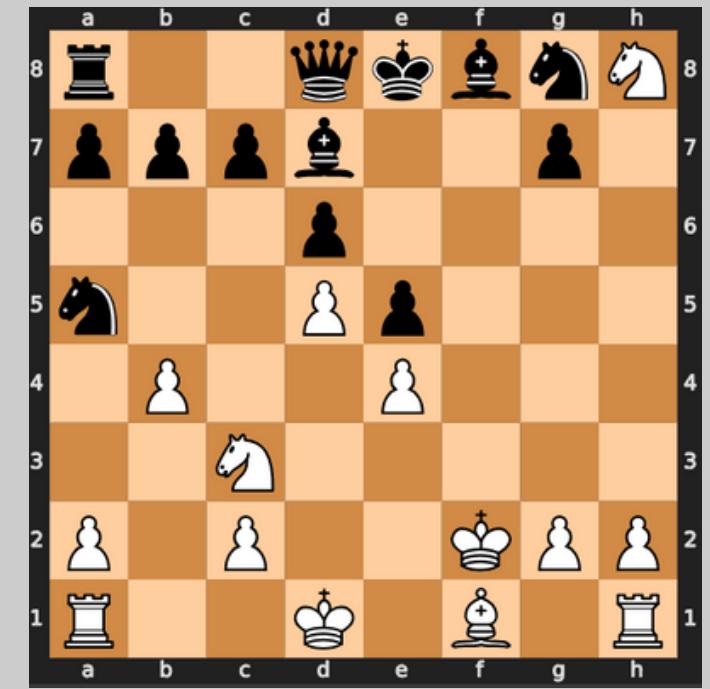
Original Image



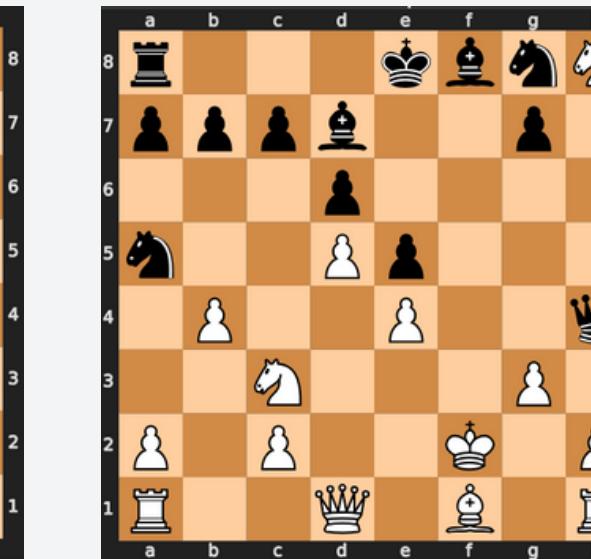
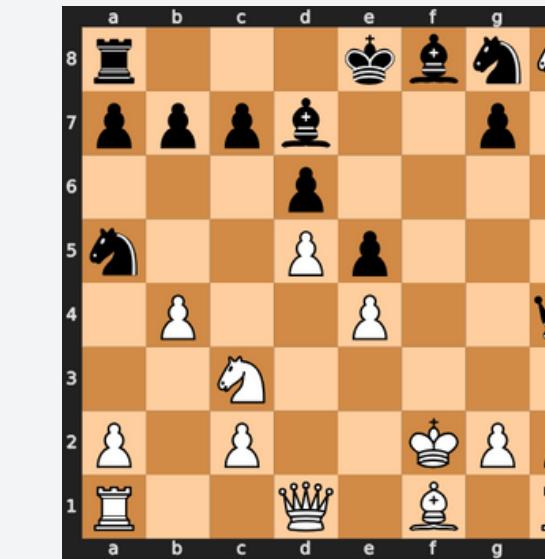
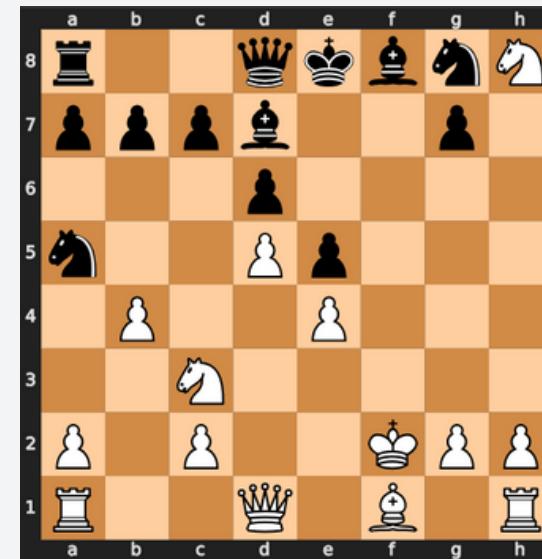
Corner Detection Model



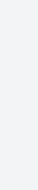
Chess Data



Convert FEN from each frame to PGN



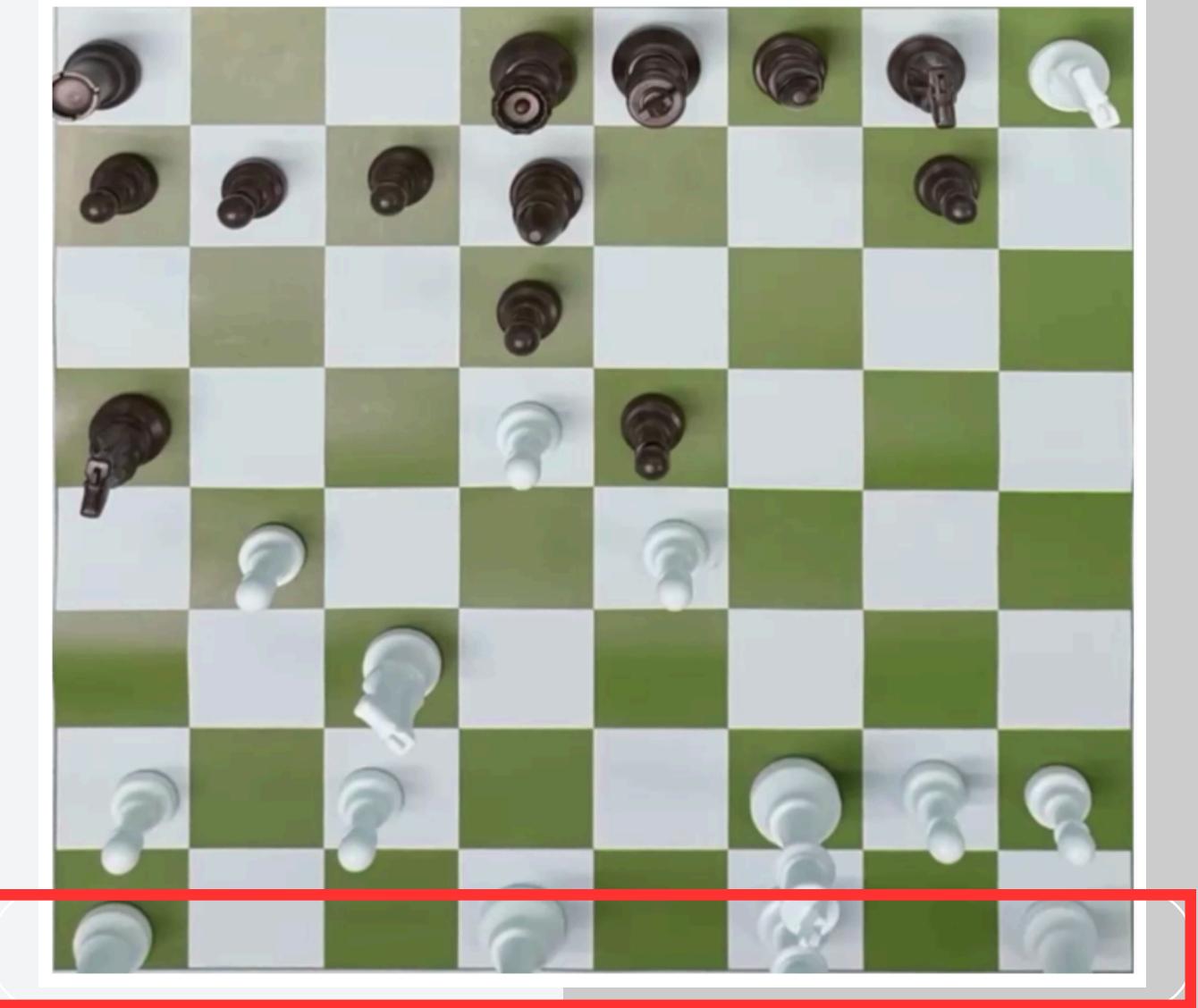
```
[ 'r2qkbnN/pppb2p1/3p4/n2Pp3/1P2P3/2N5/P1P2KPP/R2Q1B1R',
  'r3kbnN/pppb2p1/3p4/n2Pp3/1P2P2q/2N5/P1P2KPP/R2Q1B1R',
  'r3kbnN/pppb2p1/3p4/n2Pp3/1P2P2q/2N3P1/P1P2K1P/R2Q1B1R' ]
```

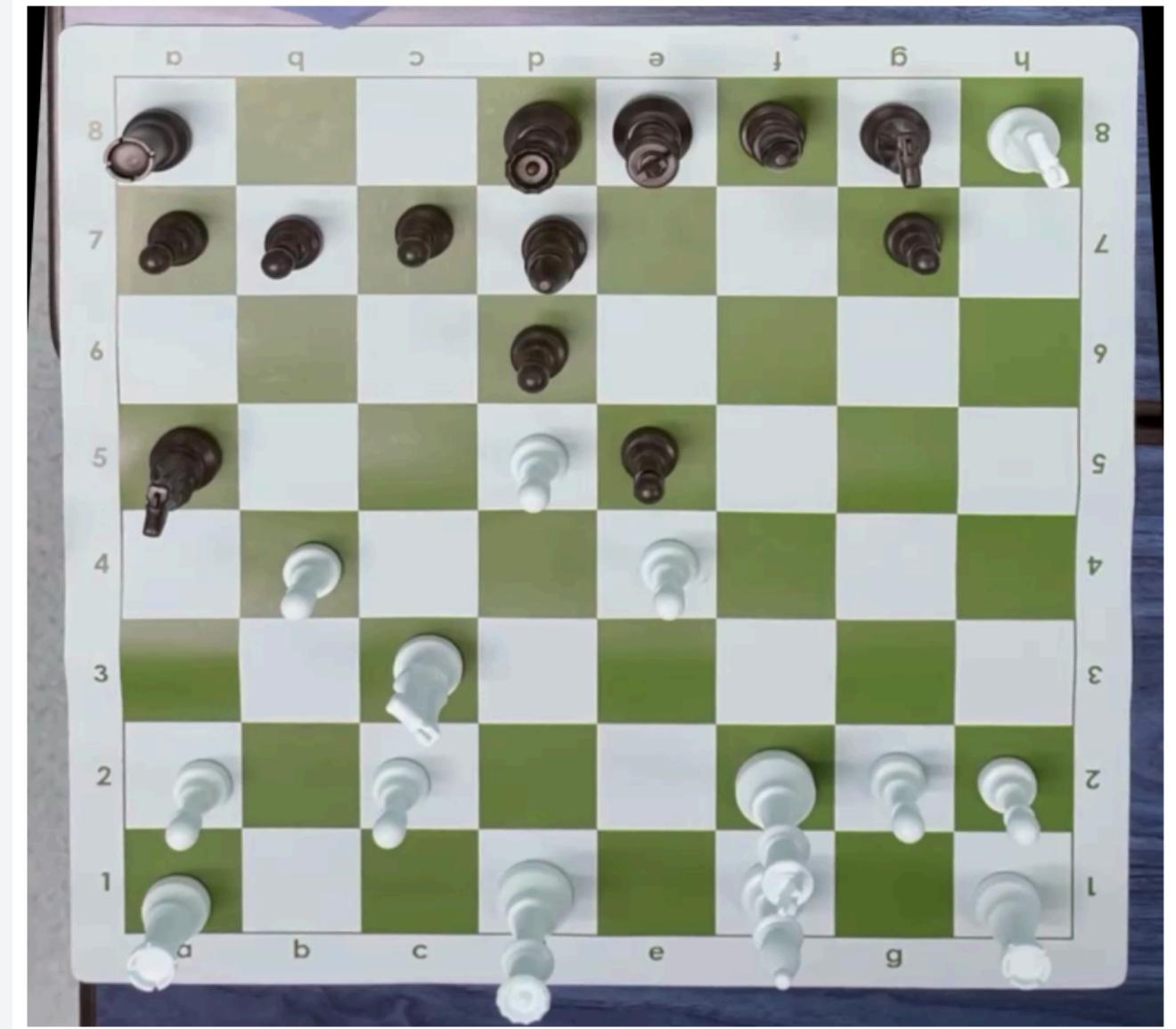


1... **0h4+** 2. **g3**



No padding





Add padding

PGN to CSV file

```
          row_id \
0  2_Move_rotate_student.mp4
1    2_move_student.mp4
2    4_Move_studet.mp4
3    6_Move_student.mp4
4    8_Move_student.mp4

          output
0
1... Qh4+ 2.g3
1... Qh4+ 2.g3
2
3... Bxb5 2.Rxb5
4 1.Qe6 2. Kd8 3.Qf7 c6 4.Qf2 5. Qxa7 6.Rc8 7...
```



CSV

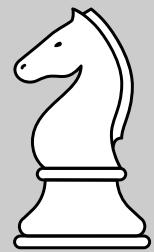
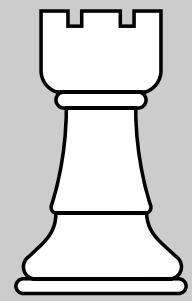
After we got PGN then, we aggregate to CSV file

Problem & Improvement

1. Models are not accurate enough

2. We tried Fast-RCNN and DETR with Roboflow datasets and fine-tuned it with in-class data, but it still couldn't outperform YOLO. Perhaps more training time is needed.

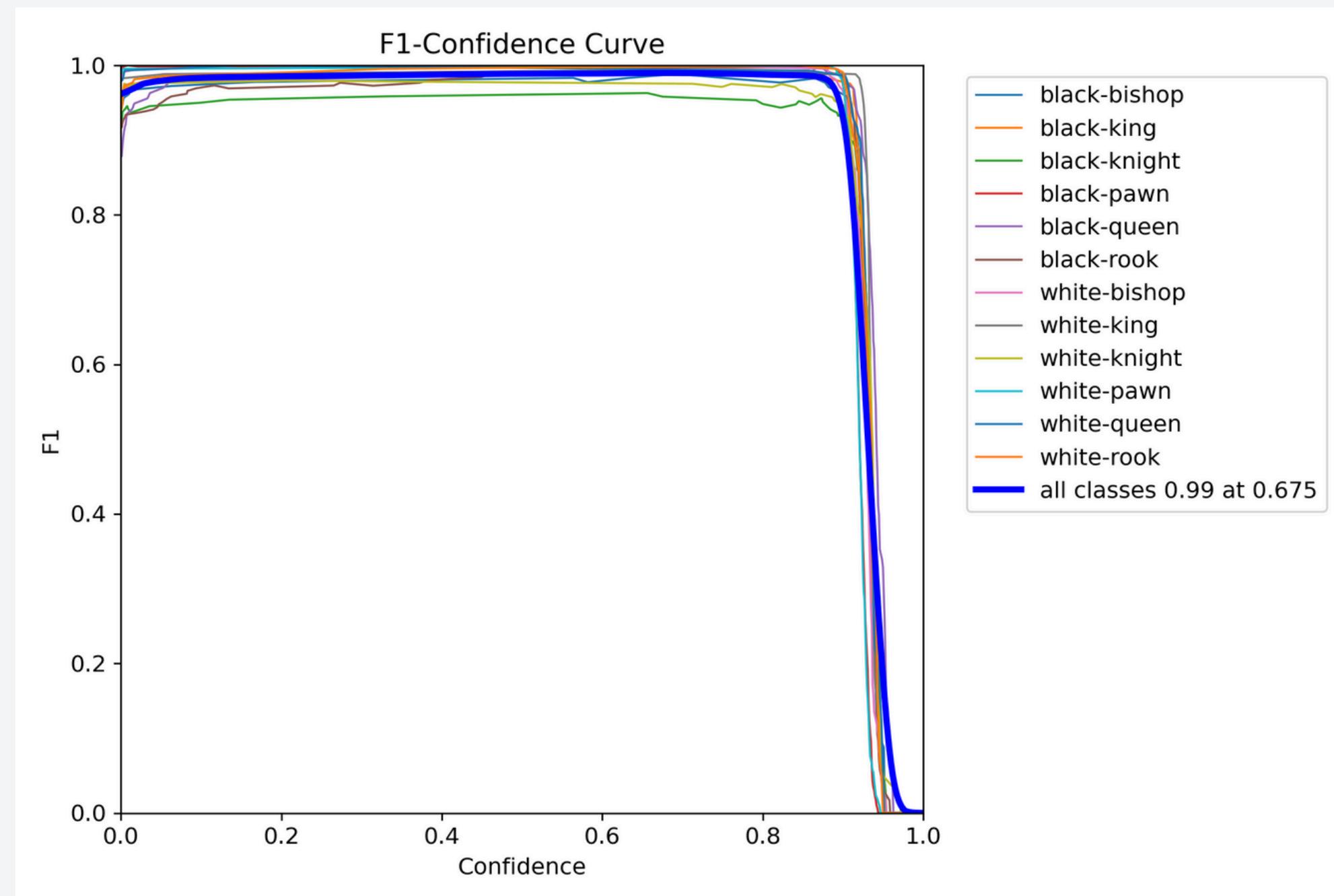
3. upgrading from YOLOv8 to YOLOv11m improved accuracy. I believe using a larger model could enhance it further.



Evaluation (chess detection)

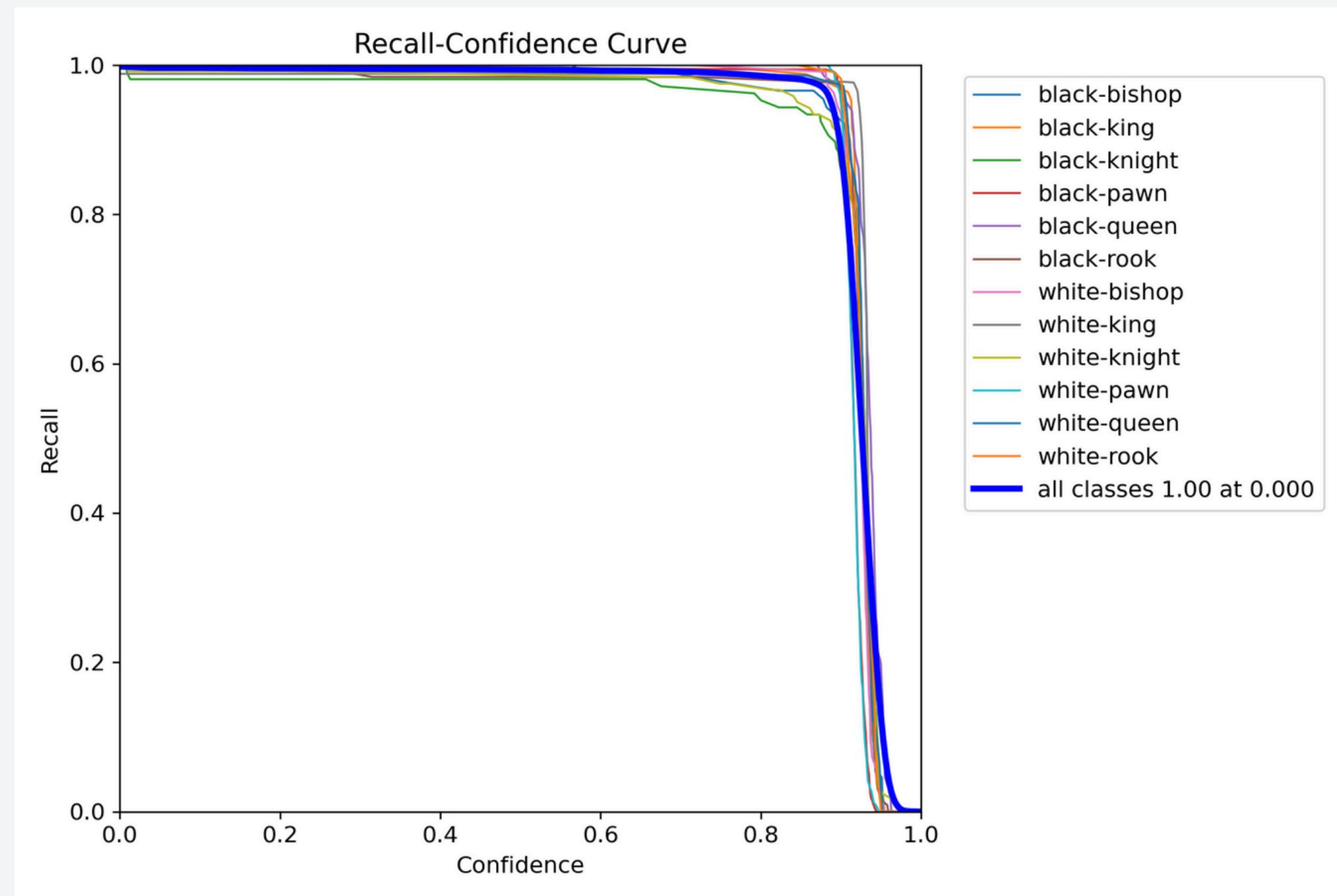
```
ultralytics 8.3.48 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv11m summary (fused): 303 layers, 20,039,284 parameters, 0 gradients, 67.7 GFLOPs
val: Scanning /content/valid/labels.cache... 216 images, 0 backgrounds, 0 corrupt: 100%|██████████| 216/216 [00:00<?, ?it/s]
      Class   Images Instances   Box(P)        R    mAP50    mAP50-95): 100%|██████████| 7/7 [00:10<00:00,
      all       216     1633     0.988     0.992    0.993     0.903
      black-bishop    54       88     0.985     0.989    0.994     0.899
      black-king      83       83     0.987         1     0.984     0.9
      black-knight     68      106     0.945     0.973    0.987     0.886
      black-pawn       76      340         1     0.995    0.995     0.882
      black-queen      55       65     0.979         1     0.995     0.911
      black-rook       77      127     0.999     0.984    0.995     0.893
      white-bishop      70      110     0.999         1     0.995     0.907
      white-king        85       86     0.999     0.988    0.993     0.933
      white-knight      81      121     0.967     0.984    0.993     0.906
      white-pawn        78      329     0.996         1     0.995     0.889
      white-queen       64       72         1     0.99     0.995     0.918
      white-rook        72      106     0.997         1     0.995     0.911
Speed: 0.4ms preprocess, 27.9ms inference, 0.0ms loss, 4.3ms postprocess per image
Results saved to runs/detect/train2
```

Matrices (chess detection)



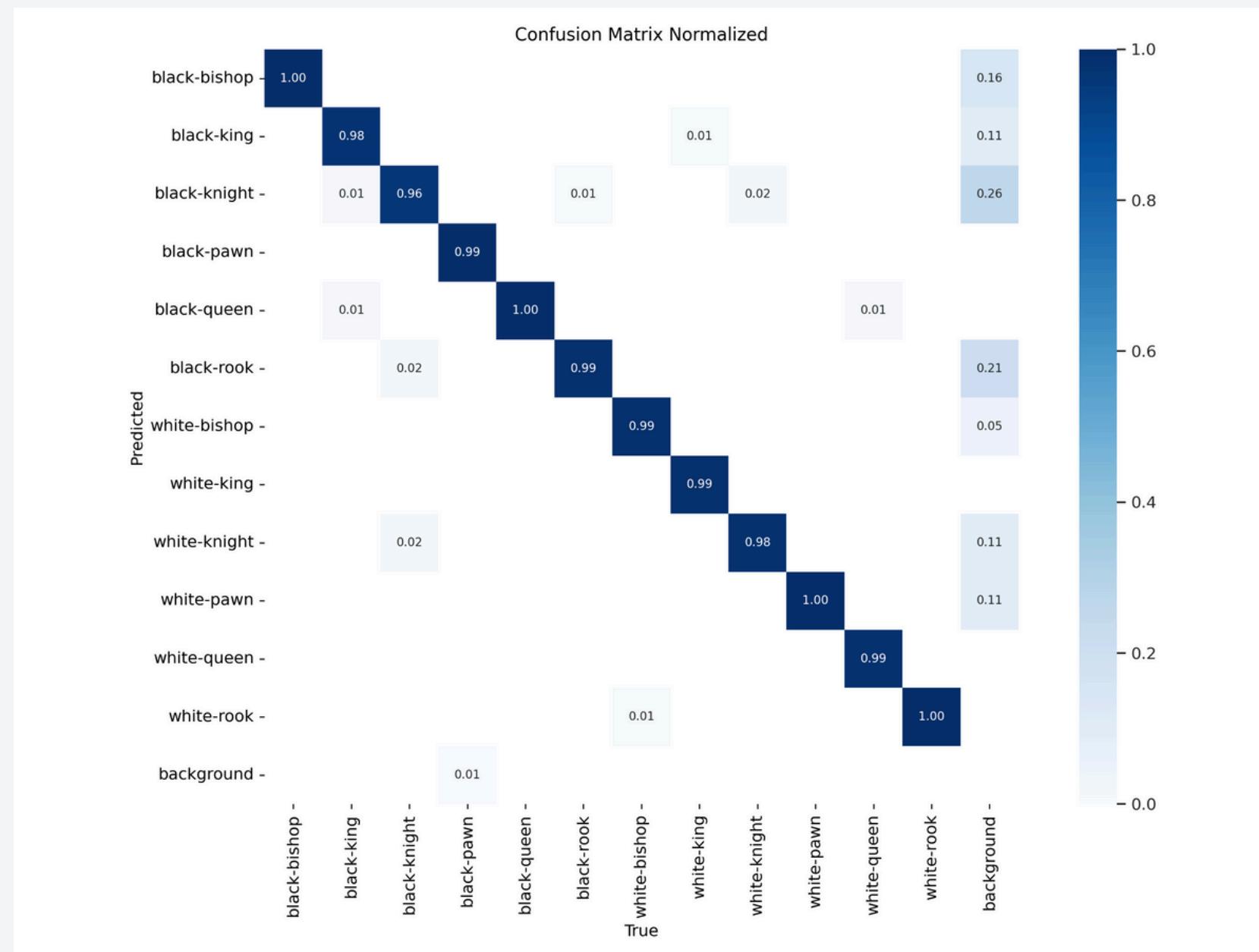
F1 curve

Matrices (chess detection)



R curve

Matrices (chess detection)



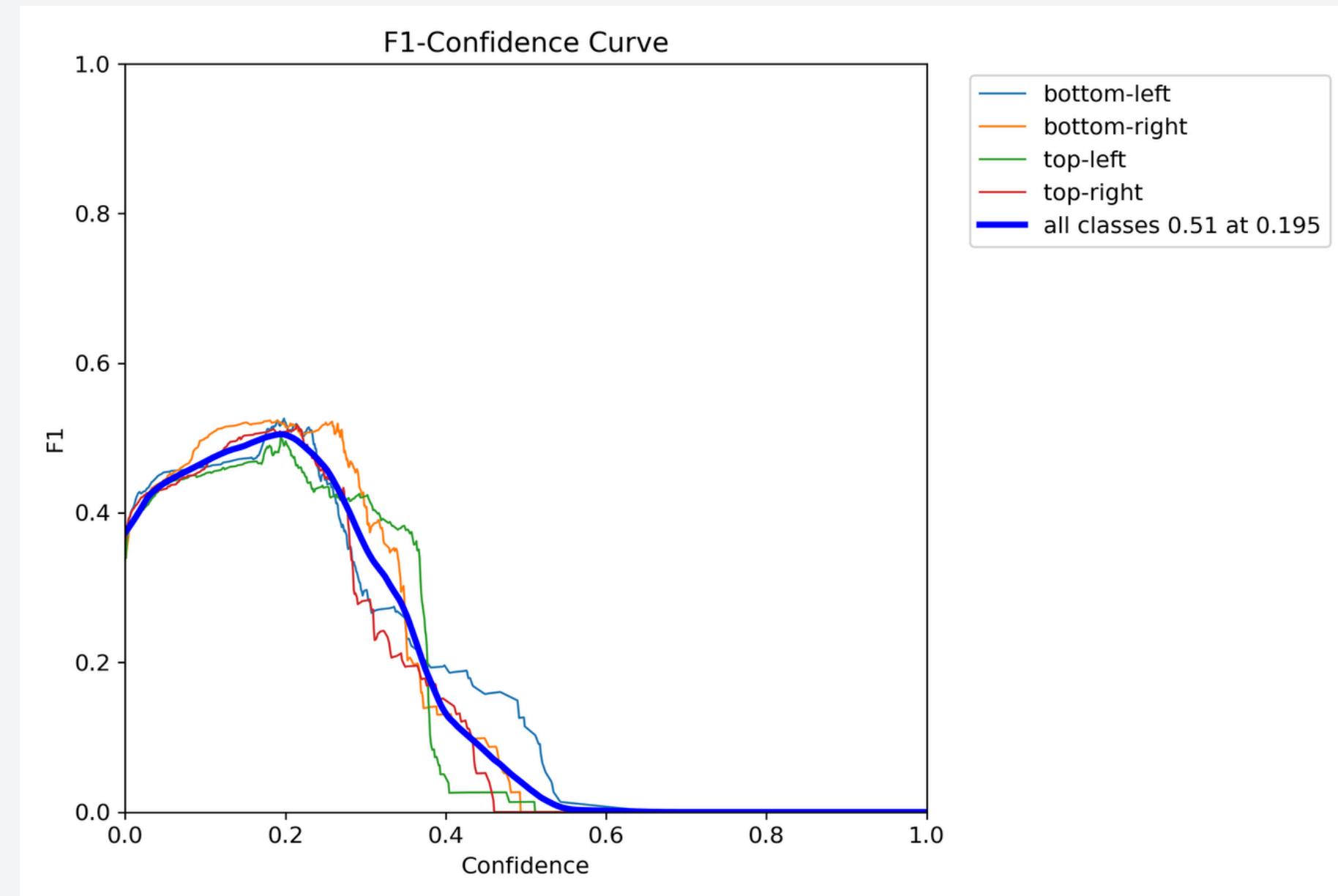
Confusion Matrix

Evaluation (corner detection)

```
Ultralytics 8.3.49 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3,006,428 parameters, 0 gradients, 8.1 GFLOPs
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 22.3MB/s]
val: Scanning /content/valid/labels... 151 images, 4 backgrounds, 0 corrupt: 100%|██████████| 151/151 [00:00<00:00, 1346.20it/s] val: New cache

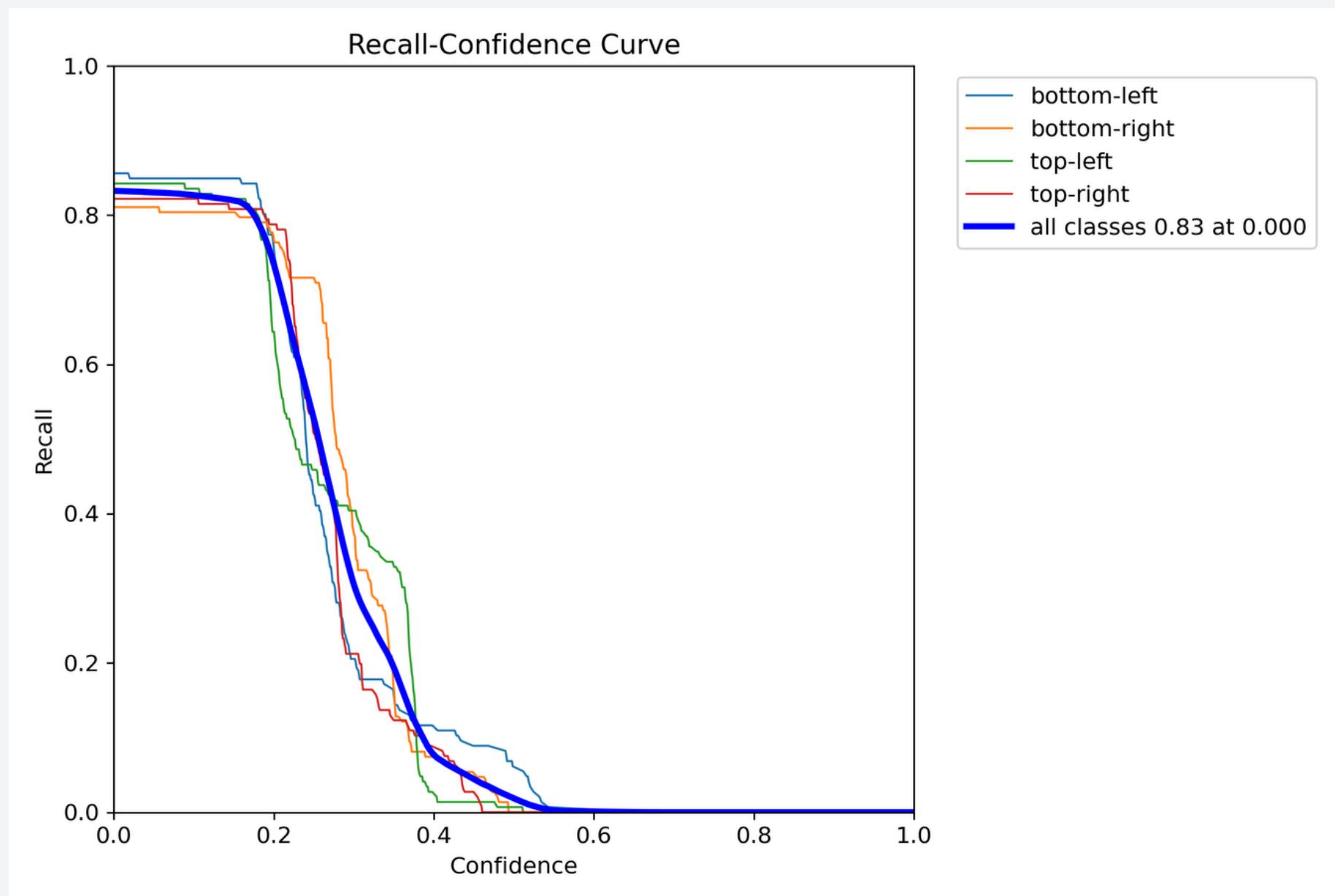
          Class    Images  Instances     Box(P)        R      mAP50  mAP50-95): 100%|██████████| 10/10 [00:04<00:00,  2.19it/s]
            all       151       586    0.371    0.809    0.407    0.165
  bottom-left     146       146    0.369    0.829    0.45    0.186
bottom-right     147       148    0.389    0.797    0.415    0.159
   top-left       146       146    0.353        0.8    0.382    0.155
   top-right       146       146    0.372    0.808    0.381    0.158
Speed: 0.9ms preprocess, 3.2ms inference, 0.0ms loss, 6.6ms postprocess per image
Results saved to runs/detect/val2
```

Matrices (corner detection)



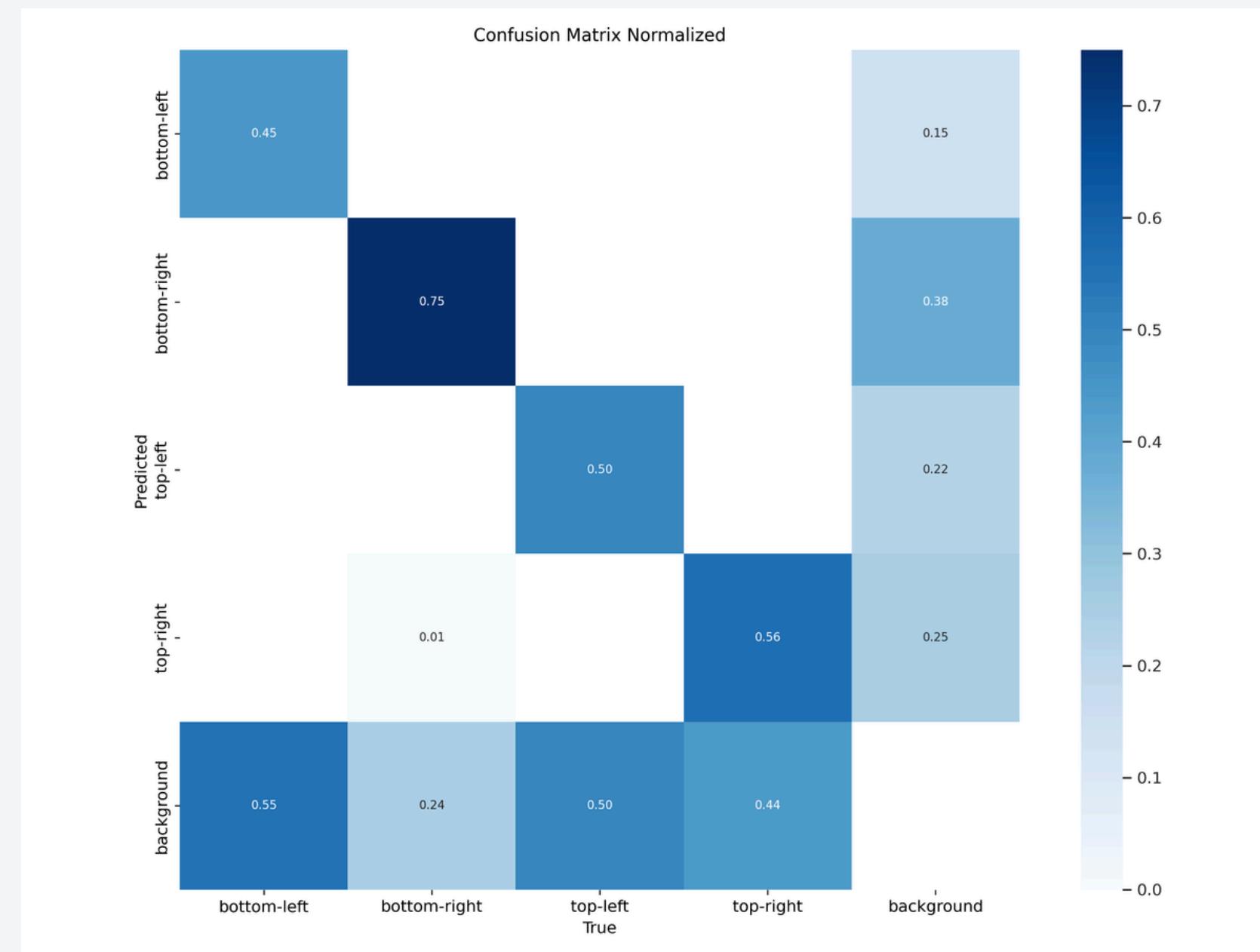
F1 curve

Matrices (corner detection)



R curve

Matrices (corner detection)



Confusion Matrix

Role

Thiraput Khongmuak

- Model
- Pipeline Design
- FEN to PGN

Chayapon Arpayatam

- Chess Detection Model
- Corner Detection Data & Model
- Evaluation Check

Chanotai Krajeam

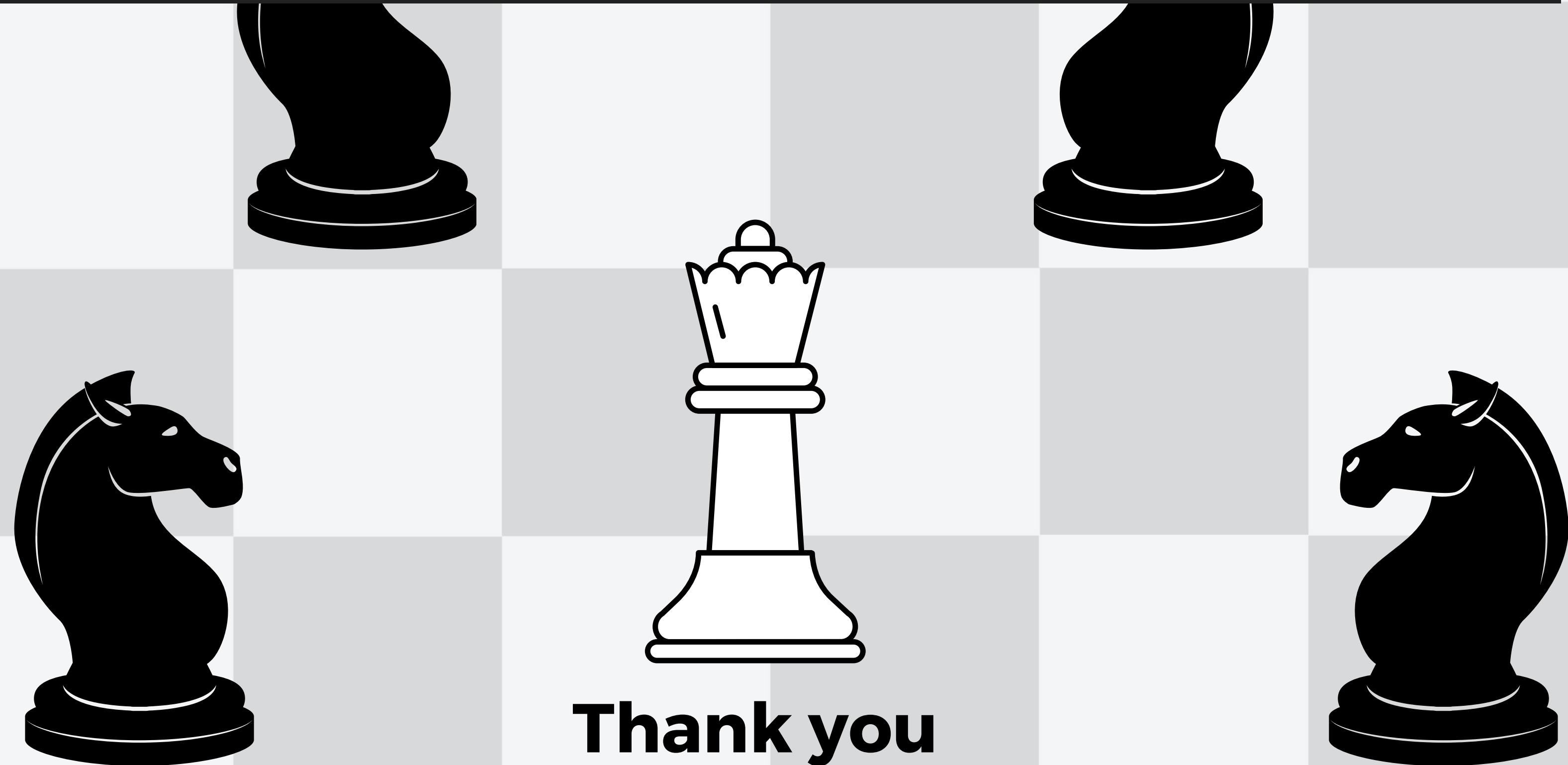
- Chess Model Detection (Fast-RCNN) (but not work)
- Corner Image Processing (Canny + Houghline) (but not work)

Chatdanai Porncharoensub

- Chess Model Detection (YOLO11m)
- Improve PGN algorithm
- Evaluation



"AI can generate images and predict chess moves. But can it predict how long I'll stare at my screen wondering what went wrong with my method?"



Thank you