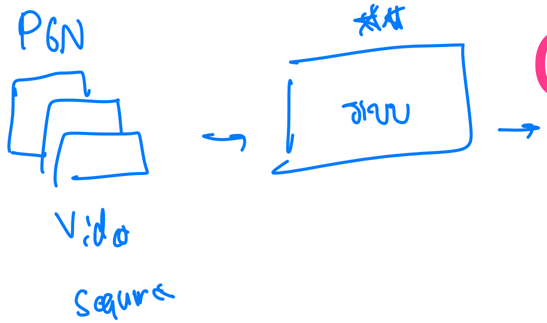
Competition: Chess Move Tracking

Chess Move Tracking

Motivation & Requirment



- Design program to detect chess piece
- Use image processing to solve this problem.

Text file

- Output Portable Game Notation (PGN) format.
- Can detect chess pieces moving each turn and another side (white - black).
- The algorithm can detect chess on video.
- Visualize its. (Optional)

Fig1. Chess board game

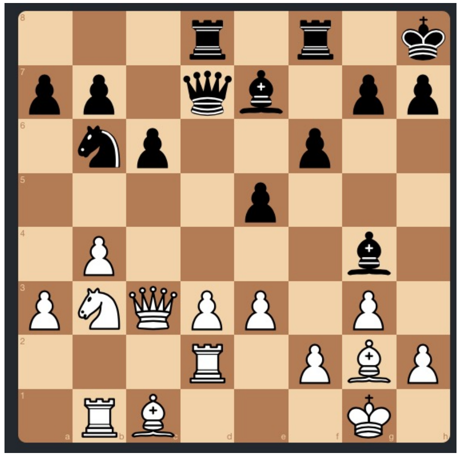


Fig2. Visualize chessboard

Visualize Chess Board Game

Chess piece notation

Chess piece notation

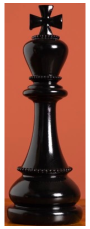


Fig1 King K

Fig2 Knight N

Fig3 Queen Q

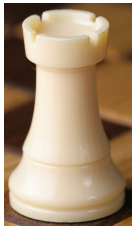


Fig4 Rook R

Fig5 Bishop B

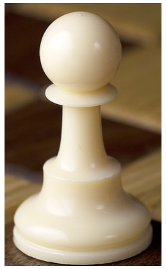
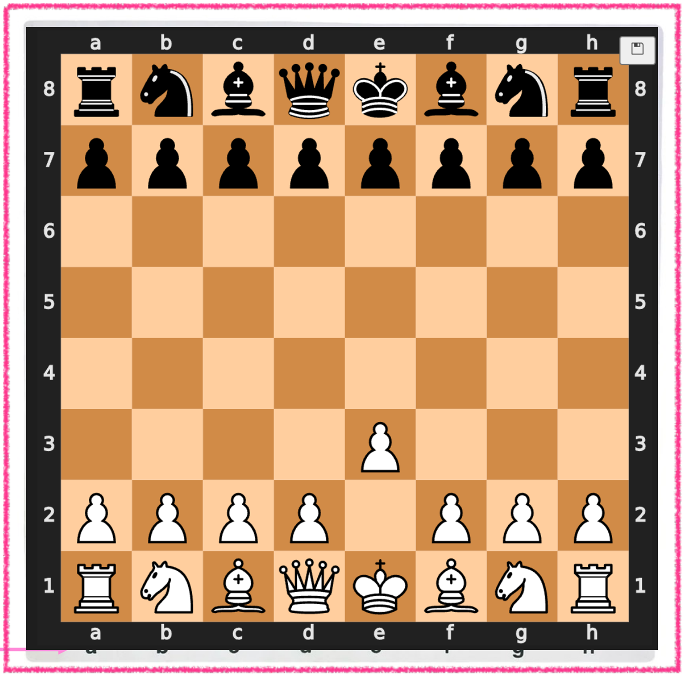


Fig6 Pawn ไม่ใส่ตัวย่อ

Chess piece direction

Chess direction Index



Dataset

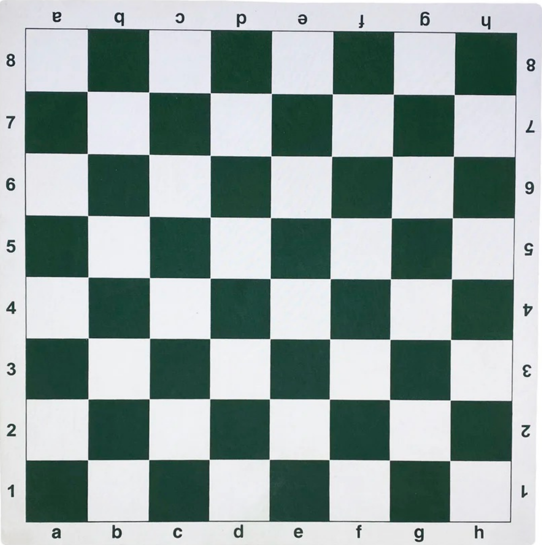
Label - PGN

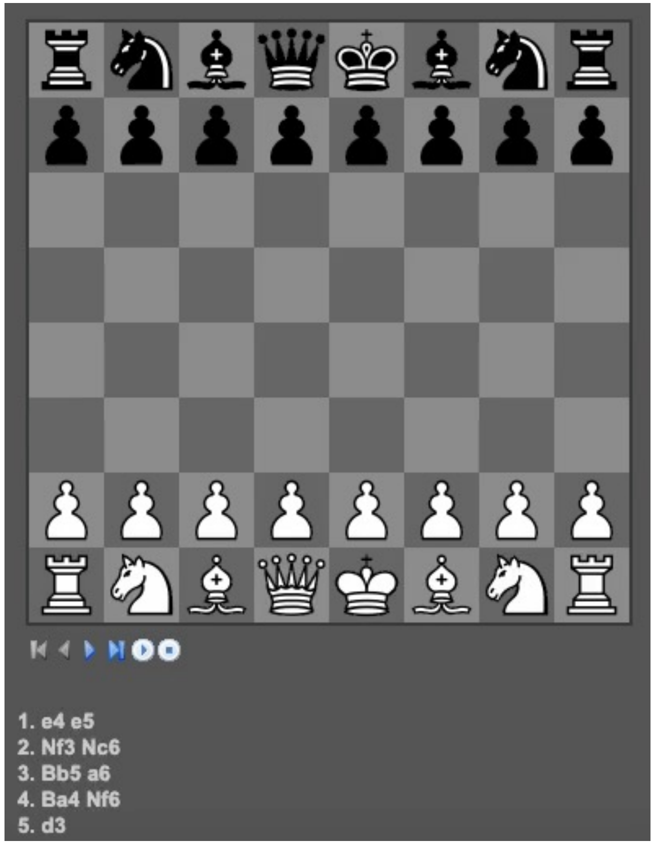
What PGN

- Standard format for recording a game in a text file
- PGN records the sequence

Why PGN

- Most chess programs support it.
- The PGN is a tool that allows players to replicate games easily using chess software





1. e4 e5
2. Nf3 Nc6
3. Bb5 a6
4. Ba4 Nf6
5. d3

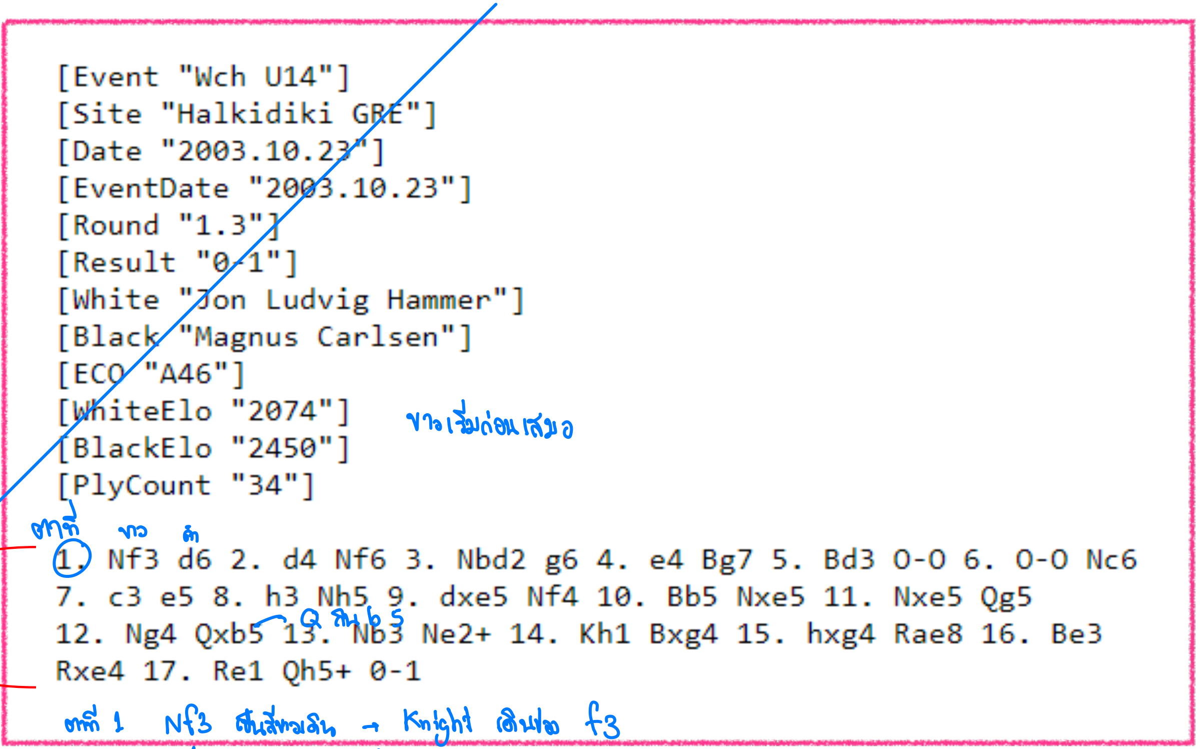
g 1. Chess game record history PGN format

Dataset

Label Video

- PGN Format
  - White always moves first, followed by Black
  - The letter abbreviations are K (king), Q (queen), R (rook), B (bishop), and N(knight). The pawn is given an empty abbreviation
  - x – capture การกินหมาก
  - See more : https://en.wikipedia.org/wiki/Portable_Game_Notation

Label - PGN



```
[Event "Wch U14"]
[Site "Halkidiki GRE"]
[Date "2003.10.23"]
[EventDate "2003.10.23"]
[Round "1.3"]
[Result "0-1"]
[White "Jon Ludvig Hammer"]
[Black "Magnus Carlsen"]
[ECO "A46"]
[WhiteElo "2074"]
[BlackElo "2450"]
[PlyCount "34"]

1. Nf3 d6 2. d4 Nf6 3. Nbd2 g6 4. e4 Bg7 5. Bd3 O-O 6. O-O Nc6
7. c3 e5 8. h3 Nh5 9. dxe5 Nf4 10. Bb5 Nxe5 11. Nxe5 Qg5
12. Ng4 Qxb5 13. Nb3 Ne2+ 14. Kh1 Bxg4 15. hxg4 Rae8 16. Be3
Rxe4 17. Re1 Qh5+ 0-1
```

ขาวเริ่มก่อนเสมอ

ตาที่ ขาว ดำ

Q กิน b5

Start scoring here

ตาที่ 1 Nf3 ฝั่งสีขาวเดิน → Knight เดินไปช่อง f3

d6 ดำเดิน → ไม่มีตัวอักษรหน้าช่อง คือ Pawn เดิน Pawn เดินไป d6

Fig 1. Example PGN Format

Dataset

Our Videos

Evaluation on Video

- Give 2 moves (Rotation)
- Give 2 moves (Original)
- Give 4 moves
- Give 6 moves + noise
- Give 8 moves

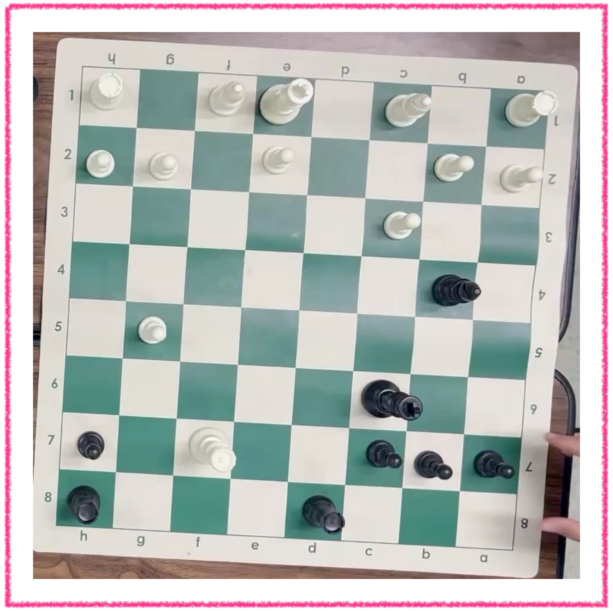


Fig 1. Example Video for 4 Moves (Rotation)

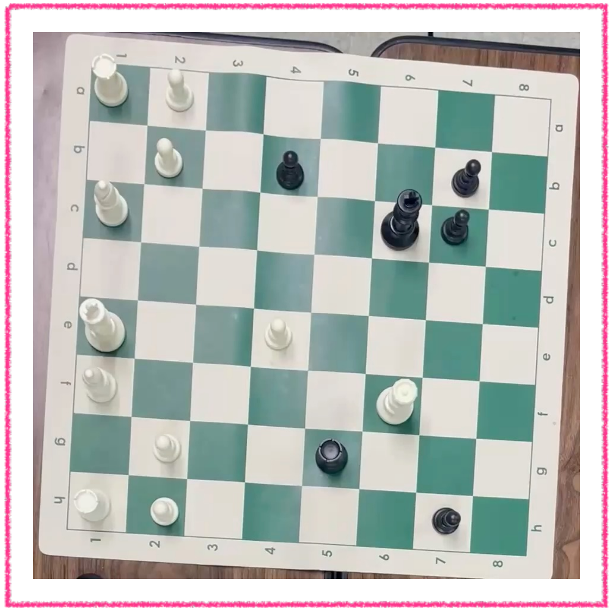


Fig 2. Example Video for 6 Moves

Link to our test data : https://chula-my.sharepoint.com/personal/6570221521_student_chula_ac_th/_layouts/15/onedrive.aspx?id=%2Fpersonal%2F6570221521%5Fstudent%5Fchula%5Fac%5Fth%2FDocuments%2FImages%5FChess%5FVideo%2FStudent&ga=1

Dataset

Public Chess piece dataset

Chess piece dataset from public on Roboflow

- Image size 426 x 416
- Raw (No Augmentation) 289 Images
- Raw + Augmentation 693 Image
- Annotation Type: Object Detection
- There are 2894 labels
- Link: https://public.roboflow.com/object-detection/chess-full

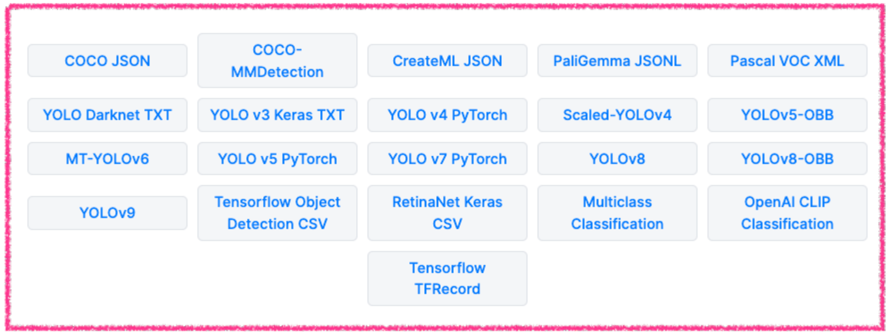


Fig 1. Available Download Formats

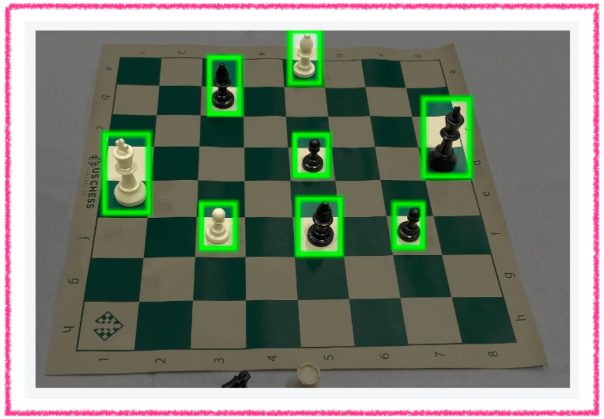


Fig 2. Example Object detection chess piece

Visualize Chess Board Game

Chess library for visualize

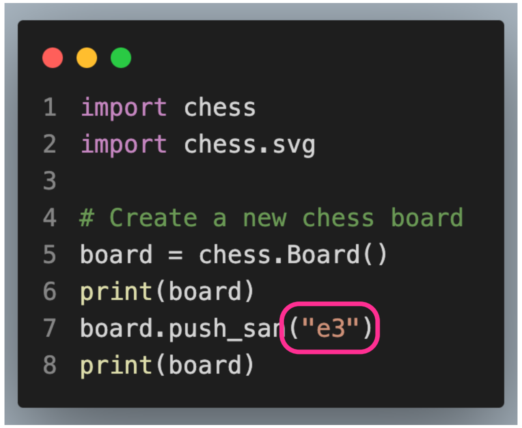
Install library

`pip install python-chess`

Example to use.

1. Init chess pieces location

2. Tell lib which pieces to move direction.

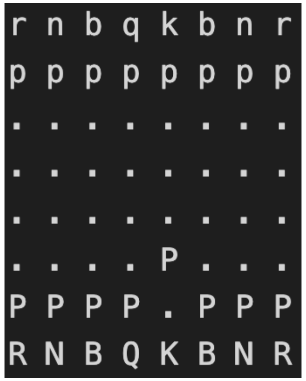


```python
import chess
import chess.svg

# Create a new chess board
board = chess.Board()
print(board)
board.push_san("e3")
print(board)
```

Fig 1. Example code to move piece

Output



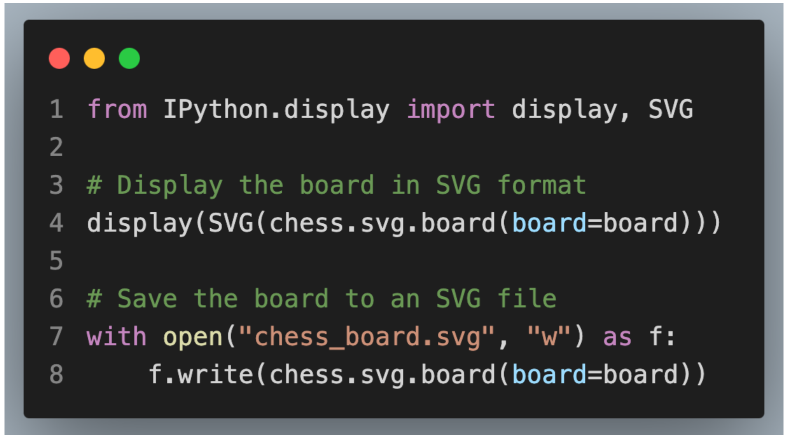
```
r n b q k b n r
p p p p p p p p
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . . P . . .
P P P P . P P P
R N B Q K B N R
```

Fig 2. Example output board from chess library

Visualize Chess Board Game

Chess library for visualize

Example to Visualize chess board to animation



```python
from IPython.display import display, SVG

# Display the board in SVG format
display(SVG(chess.svg.board(board=board)))

# Save the board to an SVG file
with open("chess_board.svg", "w") as f:
    f.write(chess.svg.board(board=board))
```

Fig1. Example code to visualize

Visualize from board

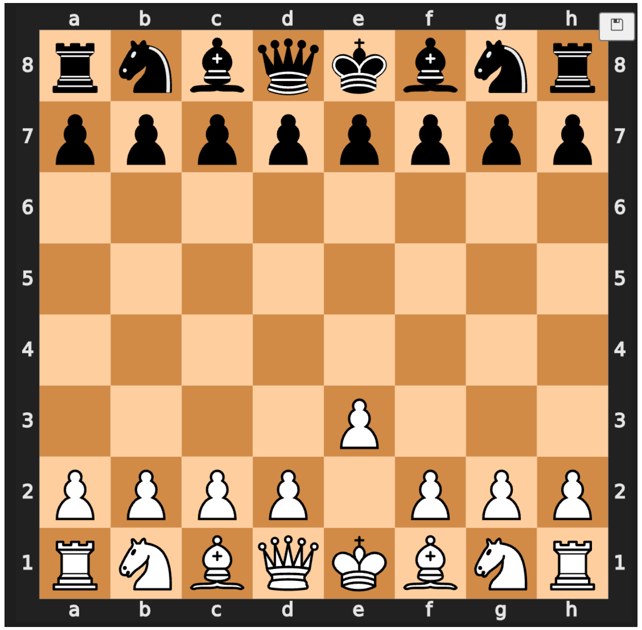


Fig2. Visualize board

Submission

- Go to https://www.kaggle.com/competitions/cu-chess-detection
- Instructions : https://uploading-prediction-out-drdxto8.gamma.site
- IMPORTANT NOTE! submissions in Kaggle! **Submission deadline: @23:59, 10 Dec 2024**
  - **Team name (same as in MCV)**
  - **Prediction file (ipynb) – Make sure it can be run properly in Kaggle**
  - **CSV file** → ไฟล์คำตอบ PGN
- MCV – TEAM NAME by Mid November
- Presentation date (Online) : Sat 14/12/2567 – 9-12 (?)

Q&A:
TA office hours will be available at the following times:
Monday: 8:00 PM - 9:00 PM Thursday: 8:00 PM - 9:00 PM During these times, please feel free to reach out to @Print and @Zeekk for any questions or assistance!

Evaluation Criteria

35 points (25%) – MAX 4 people / 1 group

1) (10 points) Image Processing/machine learning/deep learning techniques – understand and describe how you can apply it in your application

2) (5 points) Evaluation and analysis – data for testing should be varied and show the results and analyze the limitations (pros and cons) of the technique for your selected application.

3) (5 points) Identify role description of each member clearly, e.g., detailed work for pre-processing, feature extraction, deep learning model, evaluation, post-processing, solving problems, etc.

4) (5 points) Accuracy (Kaggle OR/AND Local test) – This could be fairly adjusted during the competition.

5) (5 points) Writing your idea and finding in E-poster and, make sure to include necessary information in the poster.

6) (5 points) Peer review – scoring from your friends / TAs

Presentation performance will be scored with 1) - 3)

The Winner

- Publish your code on Github
- Clean up code + Create documentation
- Winner’s Award (To be announced) - based on the output / results