Computer Vision Term Project

2025

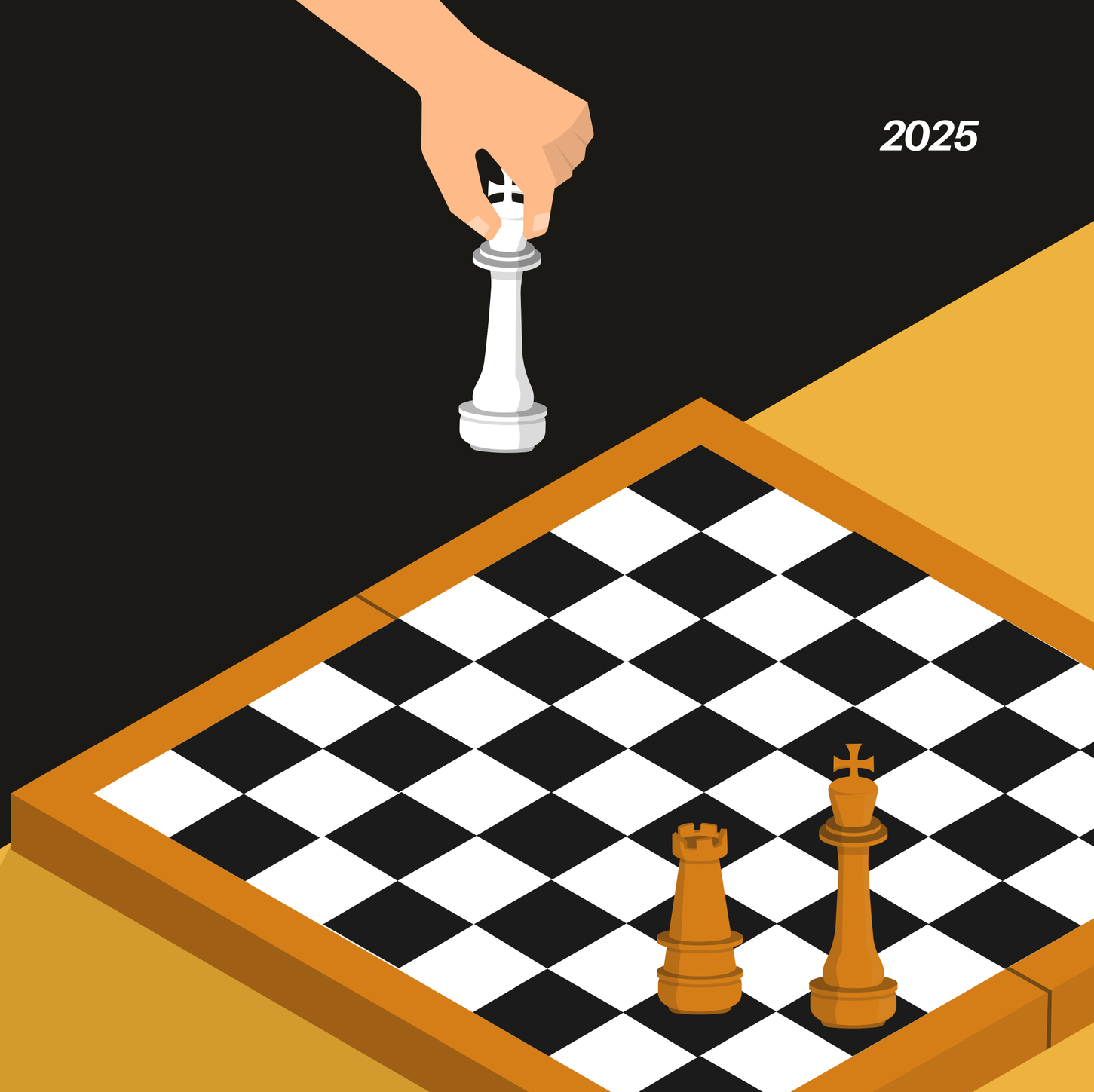
ChessLens

Team BSB64

Members

Darpan • Ronak • Soham

Ramakrishna Mission Vivekananda
Educational and Research Institute



Team BSB64

Why is Chess Fun?

Chess challenges the mind while rewarding creativity, strategy, and focus. Every move is a chance to outthink your opponent in a game where no two matches are ever the same. It's a universal language of logic and skill — pure, fair, and endlessly deep.

"Chess is the gymnasium of the mind." - Blaise Pascal

2025

Objective of the project

Piece Localization and Real-time Chess Digitization

Real-time monitor Chess Game

Piece Localization using Occupancy Detector

2D Digitization of the real-time 3D chess game

Get to Know the Chess Pieces

Queen

Combines the power of the rook and bishop — moves in all directions.

King

Moves one square in any direction; must be protected at all costs.

Rook

Travels any number of squares vertically or horizontally.

Bishop

Glides diagonally across the board, staying on the same color.

Knight

Moves in an L-shape and is the only piece that can jump over others.

Pawn

Moves forward but captures diagonally; promotes upon reaching the last rank.

Team BSB64

Team BSB64

2025

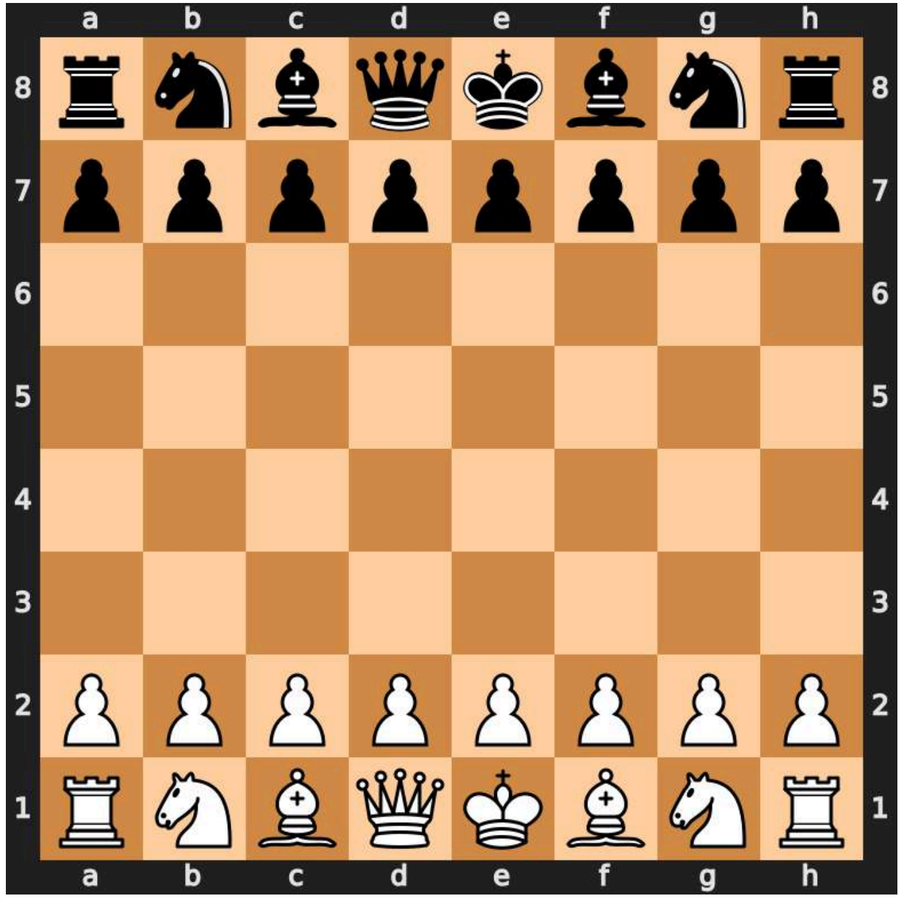
How to Play Chess?

1. *Set Up the Board: Place the board with a white square on the bottom-right; each player gets 16 pieces.*
2. *Learn the Moves: Understand how each piece moves — pawn, knight, bishop, rook, queen, and king.*
3. *Take Turns: White moves first; players alternate turns, moving one piece per turn.*
4. *Capture Opponent's Pieces: Take pieces by moving onto their square, following movement rules.*
5. *Check and Checkmate: The goal is to checkmate — trap the opponent's king with no legal escape.*
6. *End the Game: Win by checkmate, draw, stalemate, resignation, or time running out.*

Team BSB64

2025

Initial Formation



Team BSB64

2025

PLAN

- *Chessboard Detection using Homography Estimation & Perspective Warping*
- *Real-time Move Tracking*
- *Piece Localization using CNN*
- *Move Detection*
- *FEN String Generation*
- *Real-time 2D-Digitization*

Team BSB64

2025

Key Terms

1. **Homography:**
   - *A transformation that maps points from one plane to another.*
   - *Used to "warp" the perspective view of the board to a top-down view.*
2. **FEN (Forsyth-Edwards Notation):**
   - *A standard notation for describing a specific board position in chess.*
3. **CNN (Convolutional Neural Network):**
   - *A deep learning model particularly effective for image recognition tasks.*

Team BSB64

2025

Chessboard Detection using Homography Estimation

Objective

Detect the physical chessboard from a video feed and standardize the view.

1. *Extract frames from live video.*
2. *Select the four corners of the board using mouse*
3. *Apply Homography to project it to an 800x800 pixel top-down view.*
4. ***Result**: Apply Perspective Warping to get a warped image that eliminates perspective distortion.*

Team BSB64

2025

Real-time Move Tracking

Objective

Capture the frame after move has been made

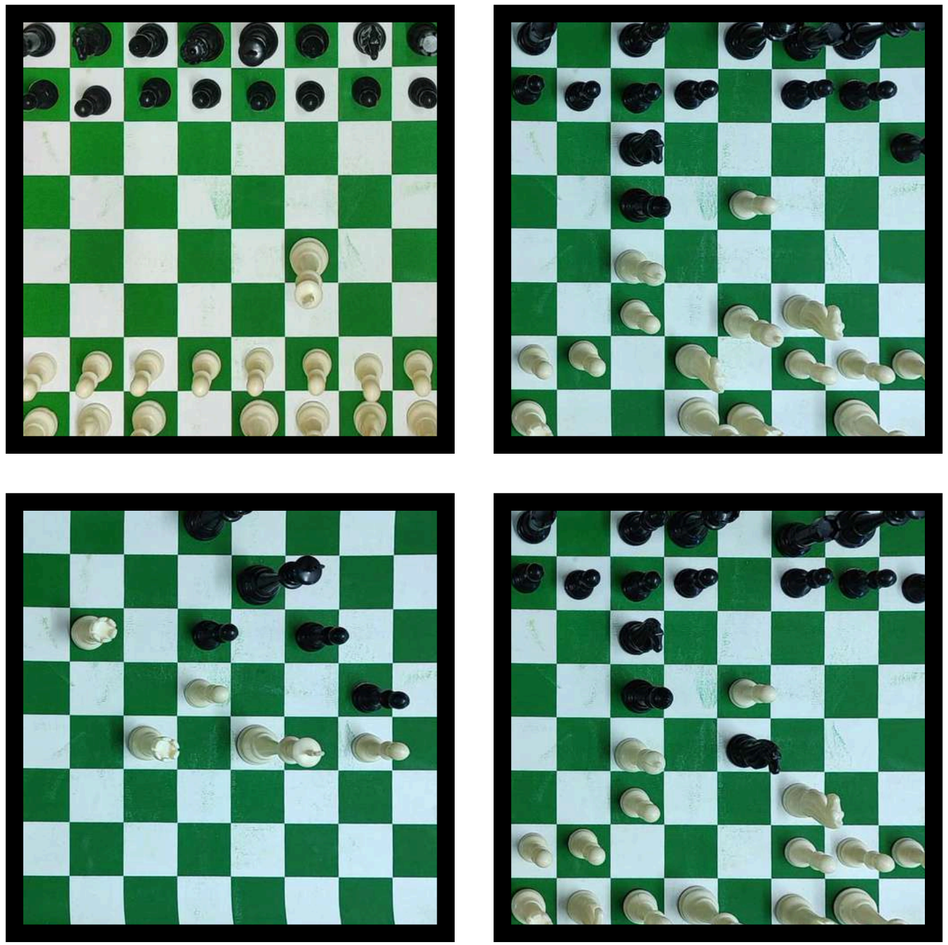
1. **Trigger Mechanism:**
   - *Players click the time-clock (here a key in the keyboard) after each move.*
   - *Captures a new frame for analysis.*

2. **Why?**
   - *Ensures frame is only processed when a move is likely to have occurred.*
   - *Reduces unnecessary computation.*

Team BSB64

2025

Dataset Acquisition

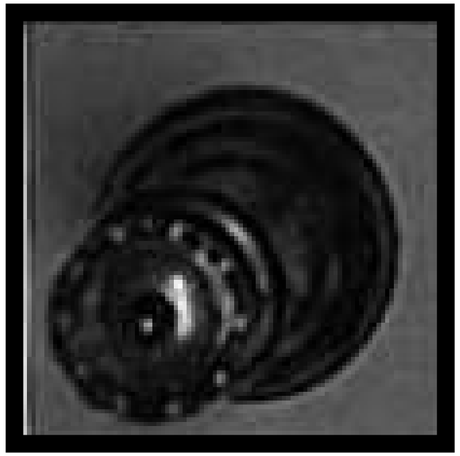


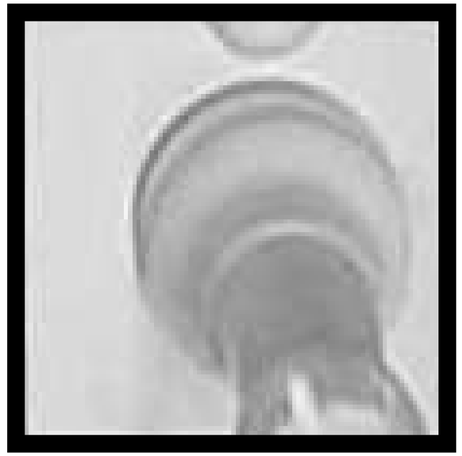
Team BSB64

2025

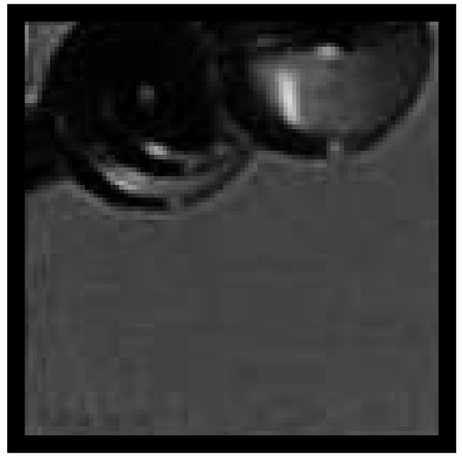
Dataset Classes

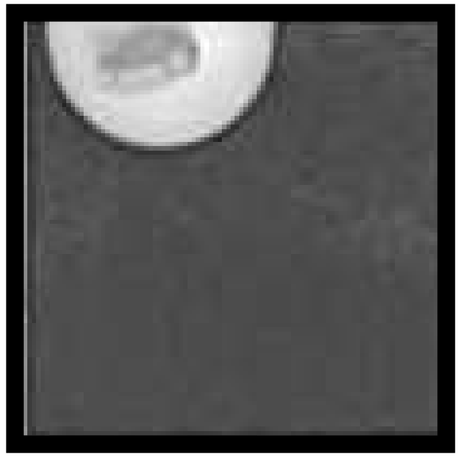
Occupied





Empty





Team BSB64

CNN Occupancy Model

2025

CNN Architecture

Input layer: 100 x 100 x 1 grayscale images

- Two convolutional layers: 32 and 64 filters, 3 x 3 kernels, ReLU activation
- Max-pooling layers: 2 x 2
- Flatten layer
- Dense layer: 64 units, ReLU activation
- Sigmoid output layer for binary classification

The model is compiled with the Adam optimizer and binary cross-entropy loss, optimized for accuracy.

Test image

Team BSB64

2025

CNN Occupancy Results

```
[16]: columns = "abcdefgh"
      s = []
      actuals, preds = [],[]
      for i in range(8):
          for j in range(8):
              box = img[i*100:(i+1)*100,j*100:(j+1)*100]
              actual = "Occupied" if (i<2)or(i>5) else "Empty"
              outcome = 1-is_occupied_cnn(box)
              predicted = "Occupied" if outcome==True else "Empty"
              s+=[f"Box ({columns[j]}{i})\t{actual}\t{predicted}"]
              actuals+=[actual]
              preds+=[predicted]

      1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
      1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
      1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
      1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
      1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
      1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
```

```
[18]: TP,TN,FP,FN = 0,0,0,0

      for i in range(64):
          a = actuals[i]
          p = preds[i]
          # print(a,p)
          TP+= 1 if (a=="Occupied" and p=="Occupied") else 0
          TN+= 1 if (a=="Empty" and p=="Empty") else 0
          FP+= 1 if (a=="Empty" and p=="Occupied") else 0
          FN+= 1 if (a=="Occupied" and p=="Empty") else 0
```

```
[21]: print(f"Accuracy : {(TP+TN)/64}")
      accuracy = (TP+TN)/64

      Accuracy : 0.9375
```

```
[22]: print(f"Precision : {(TP)/(TP+FP)}")
```

```
[22]: print(f"Precision : {(TP)/(TP+FP)}")
      precision = (TP)/(TP+FP)

      Precision : 1.0
```

```
[23]: print(f"Recall : {(TP)/(TP+FN)}")
      recall = (TP)/(TP+FN)

      Recall : 0.875
```

```
[26]: print(f"F1-score : {2/((1/precision)+(1/recall))}")
      F1 = 2/((1/precision)+(1/recall))

      F1-score : 0.9333333333333333
```

Team BSB64

2025

Piece Localization using CNN

Objective

Identify which squares are occupied by chess pieces.

1. *CNN model trained on the initial warped image of the setup board.*
2. *Detects occupancy in each of the 64 squares of current frame using occupancy detector model with an accuracy of **93.75%***
3. ***Note**: Only detects presence, not the type or color of the piece.*

Team BSB64

2025

Objective

Detect the box positions that was involved in a player's move

1. Retrieve the Occupancy Matrix of current frame– a 2D grid representing piece presence.
2. Compare the occupancy matrices of new frame with the previous one.
3. *Identify which squares changed to detect the move.*

Move Detection

Team BSB64

2025

FEN strings generation

Objective

Convert the board state into a standard format.

1. *Using the updated Occupancy Matrix:*
   - *Map it to a valid FEN string.*
   - *Example: rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR*

2. ***Why FEN?***
   Enables integration with digital chess interfaces (like Lichess, Stockfish).

Team BSB64

2025

Objective

Real-Time 2D Chess Visualization

1. Feed the FEN string into a 2D interface.
2. Digitally render the real-time position.
3. *Users can track the full game as it happens on screen.*

Real-time 2D Chess Digitization

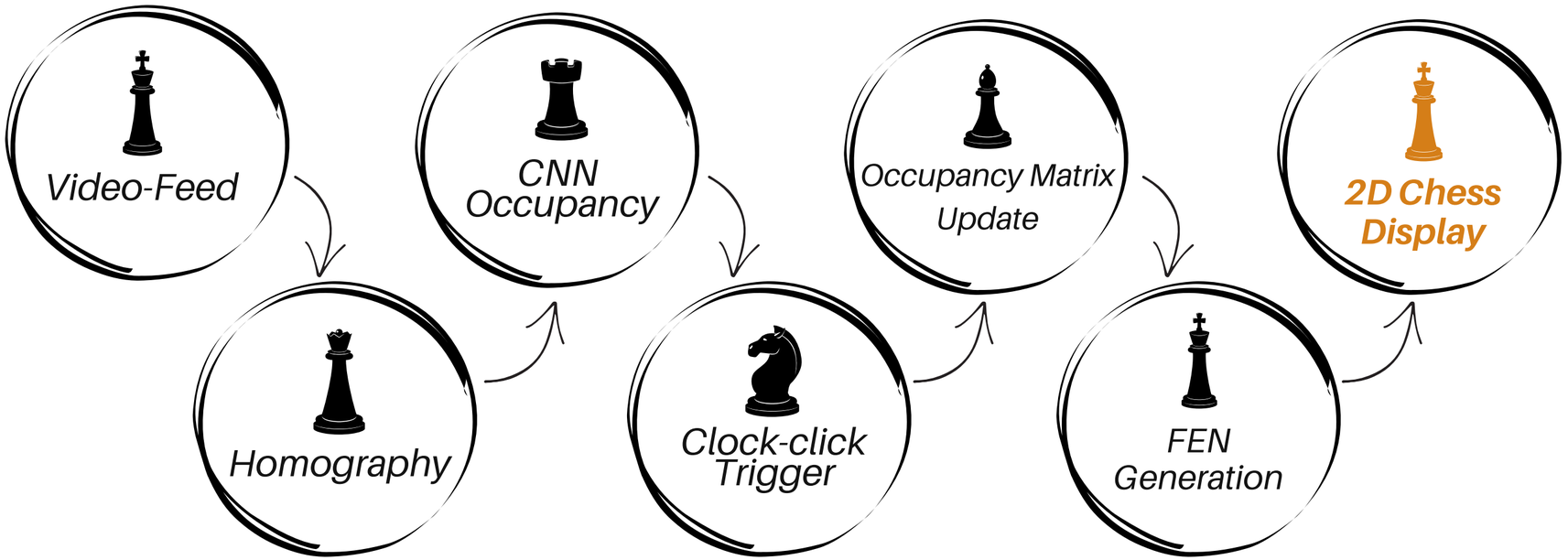
Demo Time

Team BSB64

2025

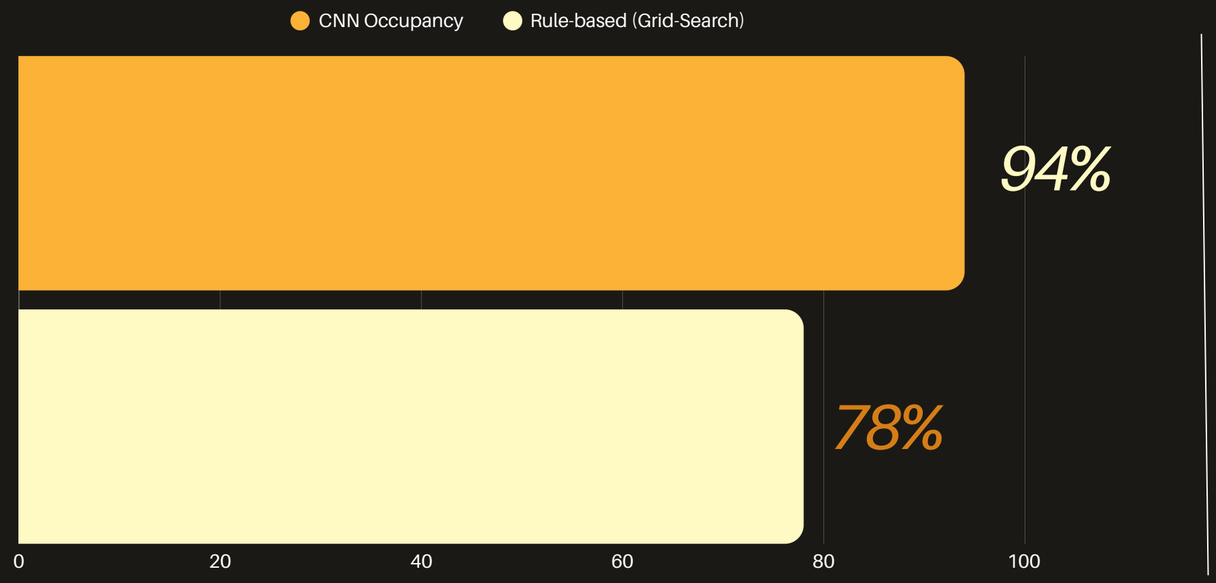
System Pipeline

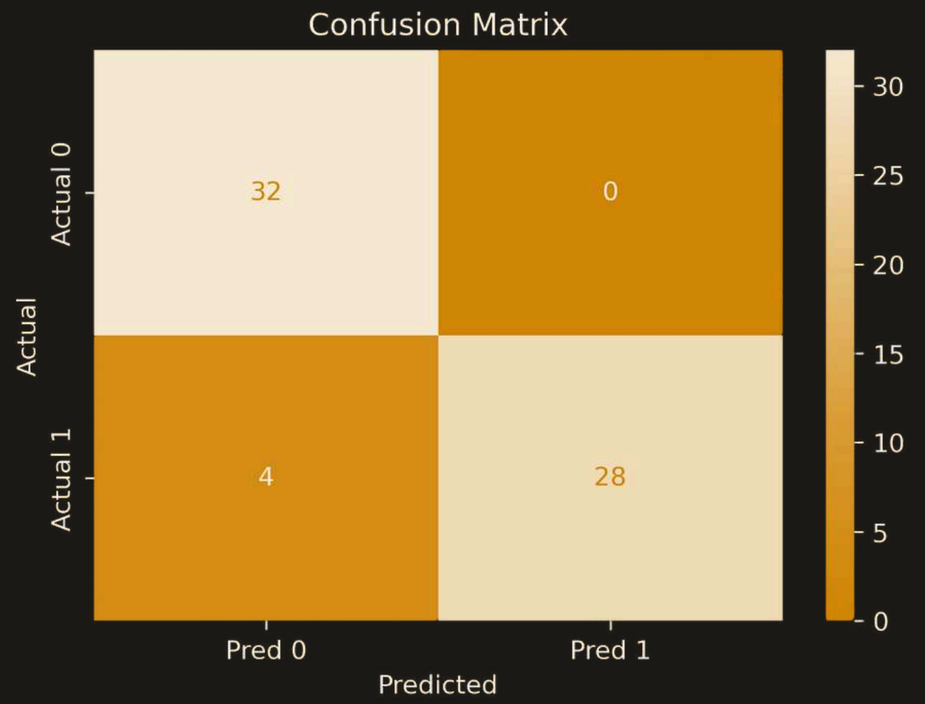
OVERVIEW



Team BSB64

Performance Results





Team BSB64

Team BSB64

2025

Future scopes

- Detect illegal moves
- Integration with chess engines for analysis
- UI enhancements and mobile deployment

Team BSB64

Thank You

"Chess is fun, challenging, and makes you smarter!"

BSB64

Darpan • Ronak • Soham

April 2025