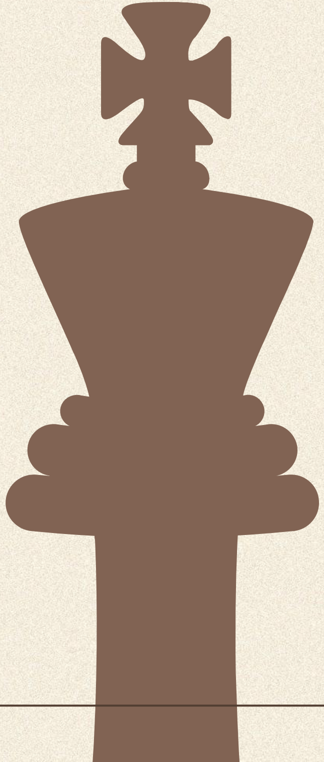
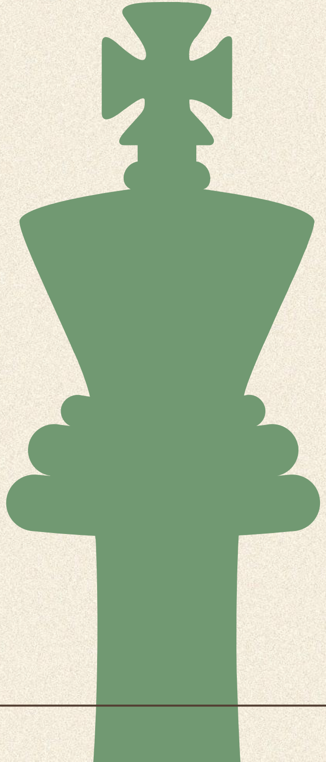
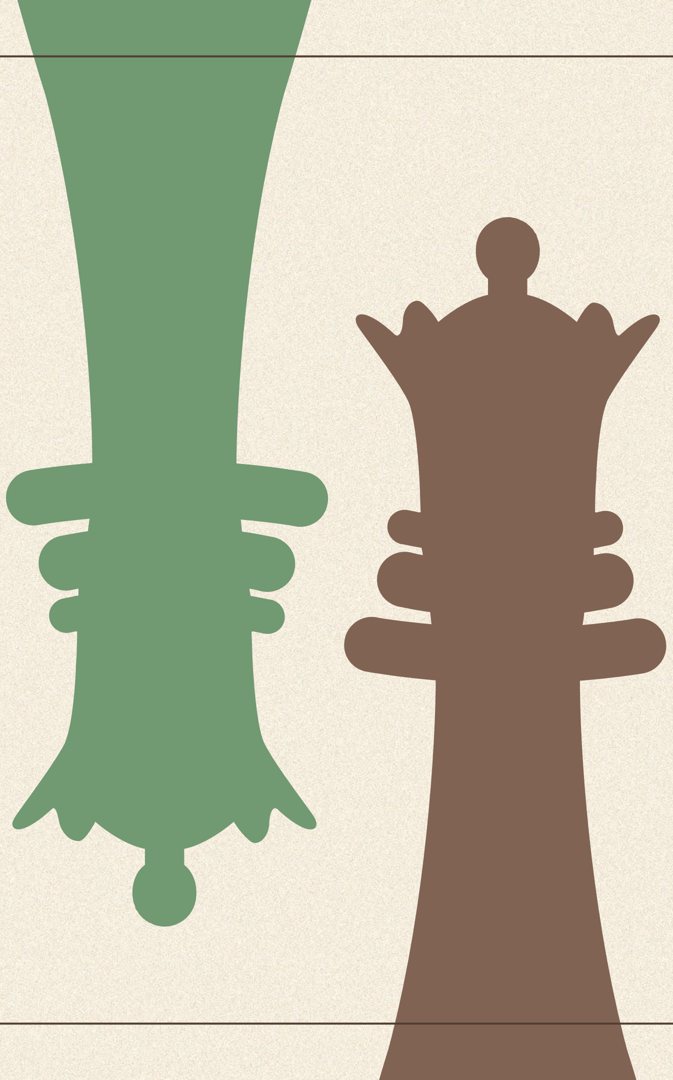


CVChess

Ved Patel, Gawthaman Senthilvelan
Luthira Abeykoon, Darshan Kasundra





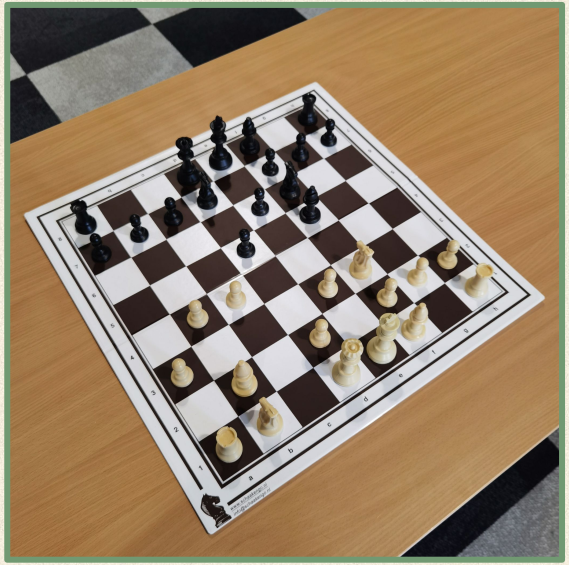
01

PROBLEM

THE CHESS RENAISSANCE

- ➢ **55% Increase** in Chess Tournament **Viewership** since 2020
- ➢ **Manually** keeping record of the moves of the game is **exhausting**
- ➢ **CVChess** proposes a way to **automate** the process

THE IDEA BEHIND CVCHESS

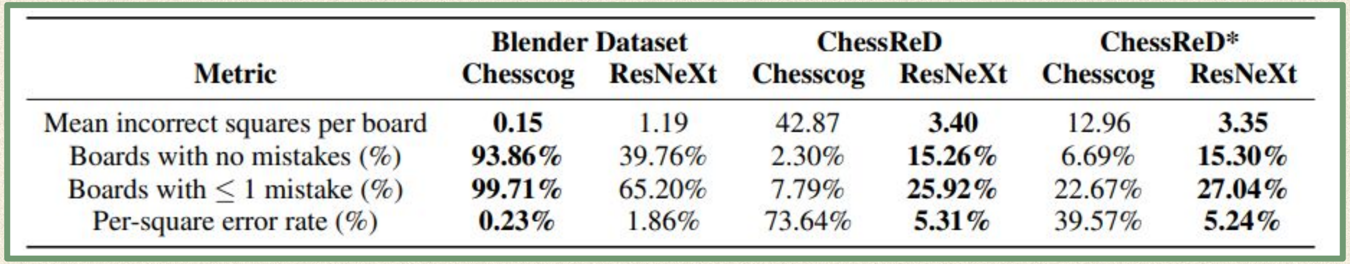


Chess Recognition Dataset (ChessReD)

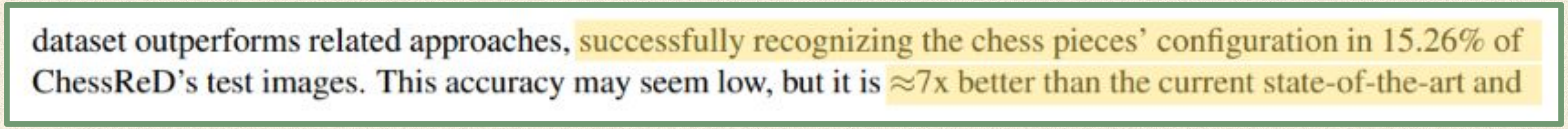
r2qkb1r/pppn1ppp/4pn2/3p1b2/1PP5/P2PN2/1B1P1PPP/RN1QKB1R

FEN Notation

THE CURRENT STATE OF THE PROBLEM

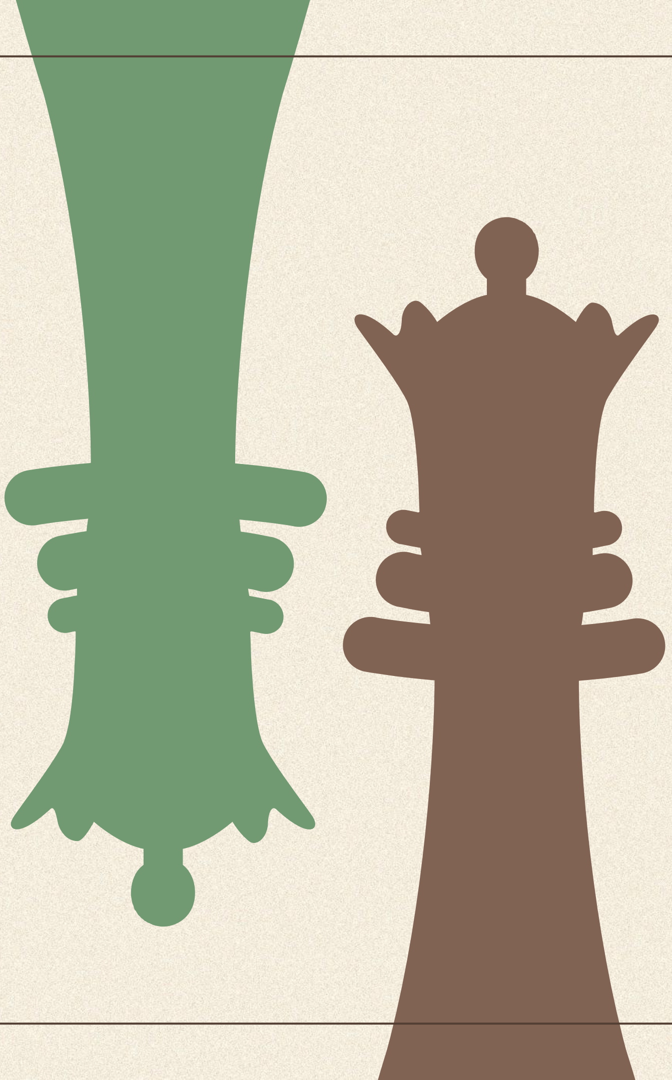


	Blender Dataset		ChessReD		ChessReD*	
Metric	Chesscog	ResNeXt	Chesscog	ResNeXt	Chesscog	ResNeXt
Mean incorrect squares per board	**0.15**	1.19	42.87	**3.40**	12.96	**3.35**
Boards with no mistakes (%)	**93.86%**	39.76%	2.30%	**15.26%**	6.69%	**15.30%**
Boards with $\leq$ 1 mistake (%)	**99.71%**	65.20%	7.79%	**25.92%**	22.67%	**27.04%**
Per-square error rate (%)	**0.23%**	1.86%	73.64%	**5.31%**	39.57%	**5.24%**



dataset outperforms related approaches, successfully recognizing the chess pieces' configuration in 15.26% of ChessReD's test images. This accuracy may seem low, but it is ≈7x better than the current state-of-the-art and

Excerpts from *End-to-End Chess Recognition by Masouris et al.*



02

DATA PROCESSING

DATASETS USED FOR CVCHESS

ChessReD

10,800 images
Various angles
60/20/20 split

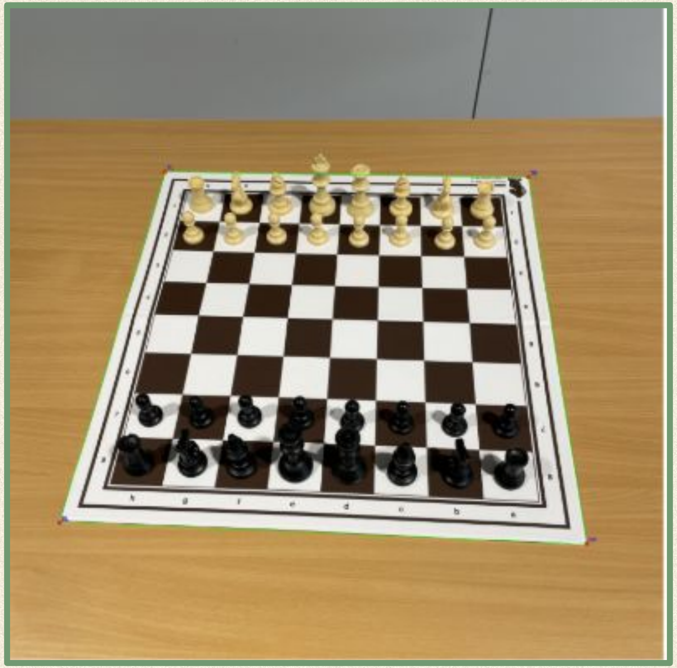
CVChess

445 images
89 unique moves
Real World Testing

BOARD DETECTION AND PERSPECTIVE TRANSFORM

- Convert to **grayscale** → **Gaussian blur** (5×5) → **Canny edges** (50–150) → **Dilate**
- Extract **largest 4-corner contour** covering >5% of image
- **Arrange corners** so a8 is top-left, h1 is bottom-right
- Apply **cv2.getPerspectiveTransform()** to warp board to a 400×400 **top-down view**

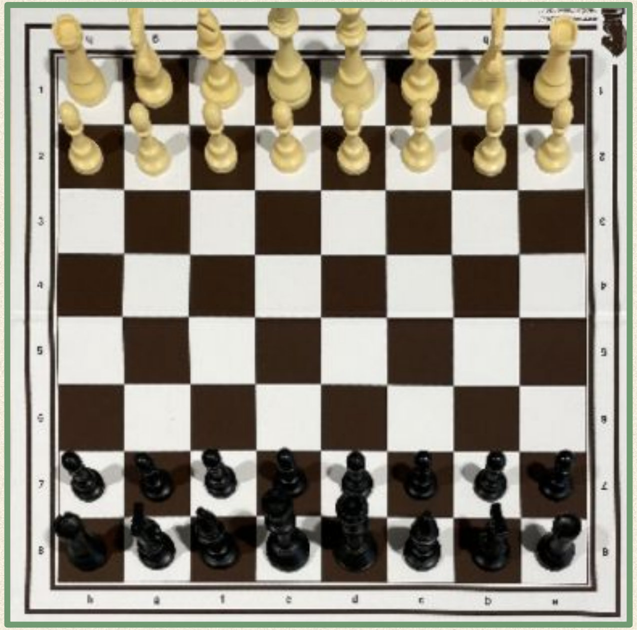
BOARD DETECTION AND PERSPECTIVE TRANSFORM



BOARD DETECTION AND PERSPECTIVE TRANSFORM

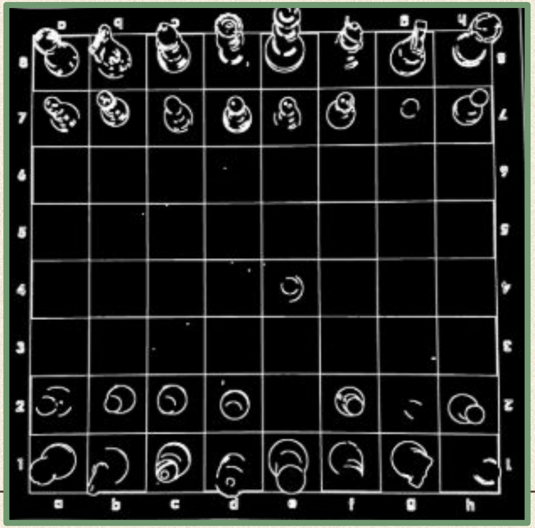
- Convert to **grayscale** → **Gaussian blur** (5×5) → **Canny edges** (50–150) → **Dilate**
- Extract **largest 4-corner contour** covering >5% of image
- **Arrange corners** so a8 is top-left, h1 is bottom-right
- Apply **cv2.getPerspectiveTransform()** to warp board to a 400×400 **top-down view**

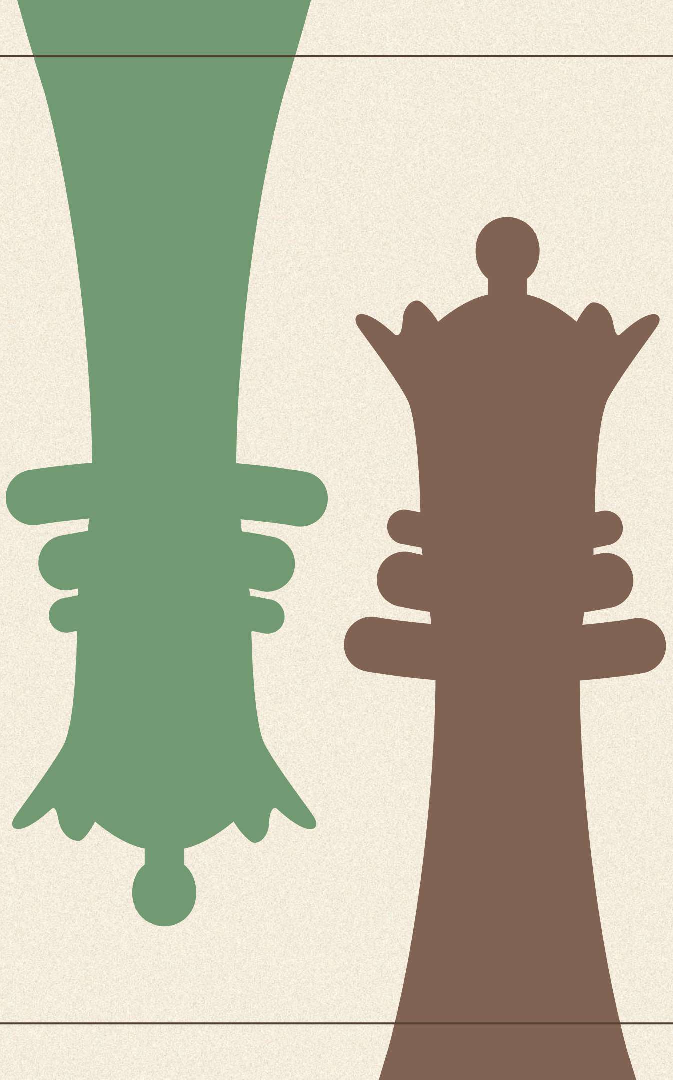
BOARD DETECTION AND PERSPECTIVE TRANSFORM



FEATURE EXTRACTION

- Corrects minor distortions and ensures accurate square boundaries
- Find line intersections → segment board into exact **8×8 grid**.
- Parse ground truth FEN → expand digits into empty squares → map each piece to **13 classes**





03

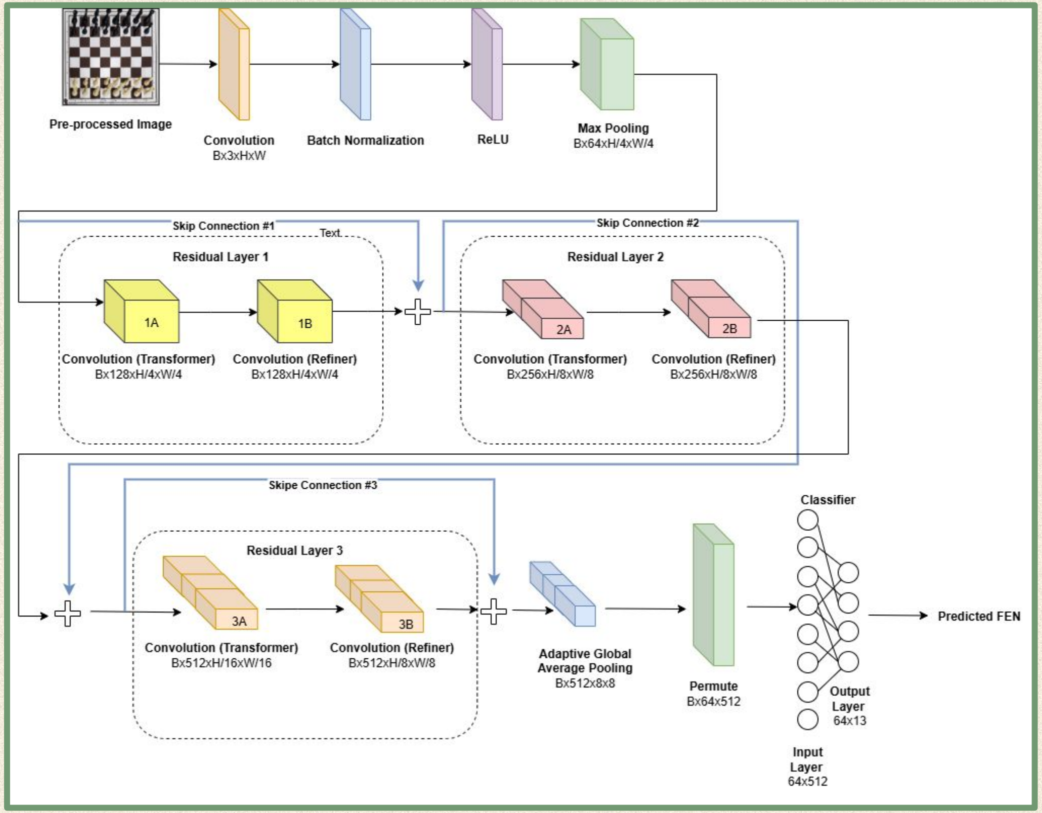
MODEL

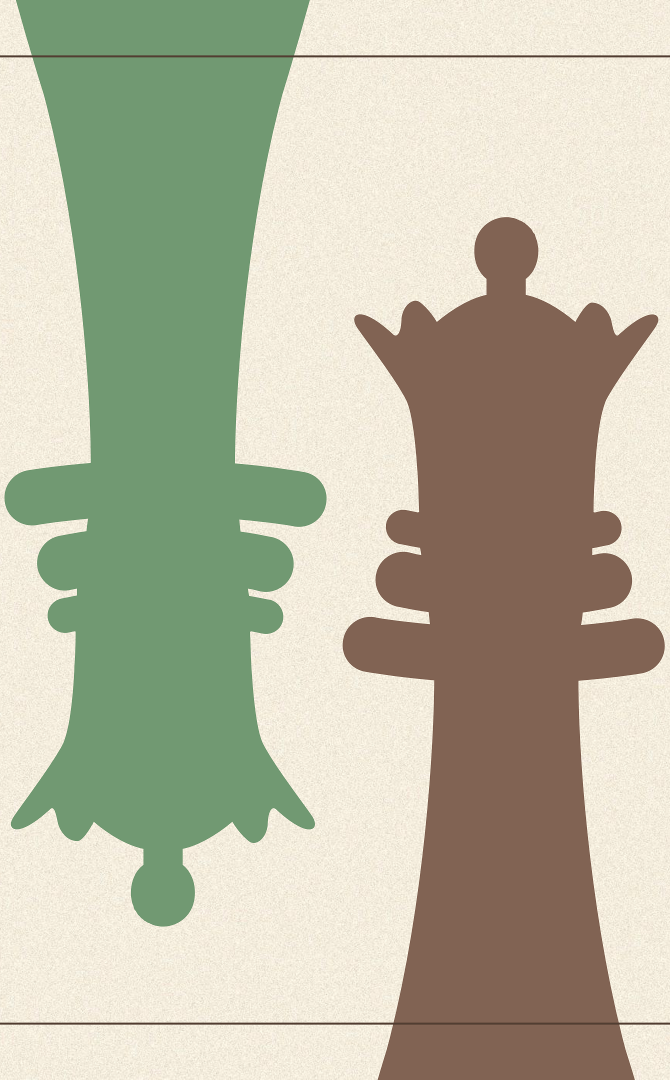
CVCHESS CNN ARCHITECTURE

- **4 Convolutional Blocks**
  - Conv2D → BatchNorm → ReLU → Pooling
  - **Input resolution:** 244x244 → progressively reduced to 8x8 (matches chess board)
- **Flatten** → Dense Layer for Classification
  - Shape of (64,13) for per-square classifying
- Predicts **1 of 13 classes** (12 pieces + empty)
- Total # of Trainable Parameters: **11,032,525**
- Design Choices:
  - **AdaptiveAvgPool** ensures spatial output matches chessboard grid size regardless of input size
  - **BatchNorm** to prevent covariate shift issues → faster convergence
  - **Dropout** (0.3) to prevent overfitting

- **Stem:** 7×7 Conv → BN → Re
  - **3 Residual Layers (Pr progression: 64→128**
  - **Input resolution:** 24
    (matches chess board)
- **Flatten** → Dense Layer for C
  - Shape of (64,13) for per
- Predicts **1 of 13 classes** (12
- Total # of Trainable Paramete
- Design Choices:
  - **AdaptiveAvgPool** ens
    grid size regardless of
  - **BatchNorm** to prevent
    convergence
  - **Dropout** (0.3) to preve

CVCHESS MODEL ARCHITECTURE

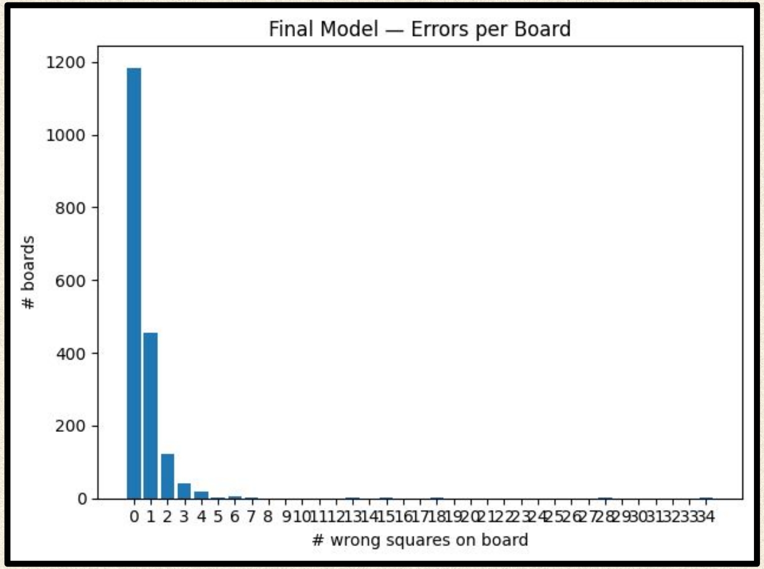




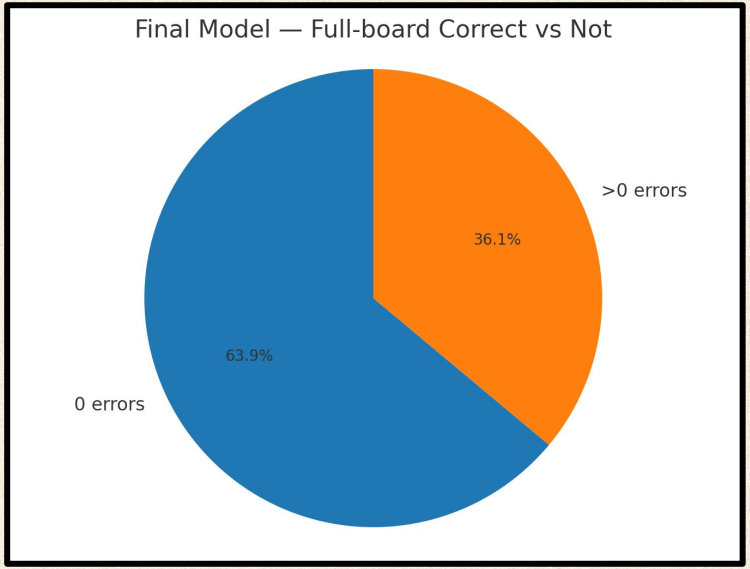
04

RESULTS

QUANTITATIVE RESULTS



Frequency of Errors Per Board

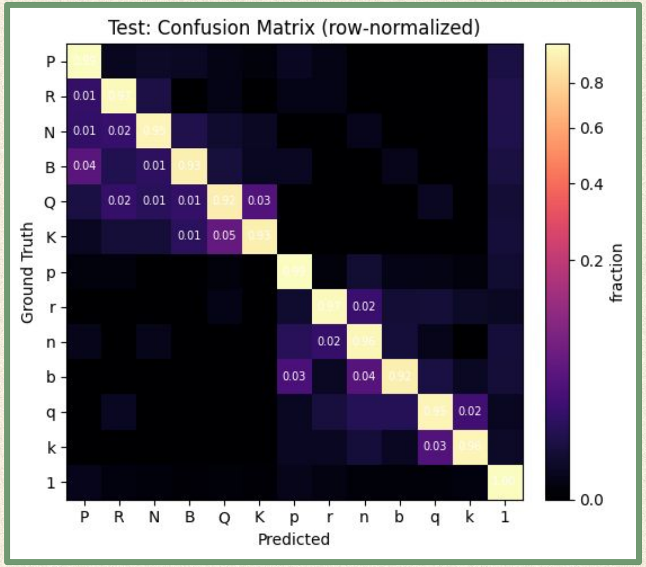


Final Model Accuracy

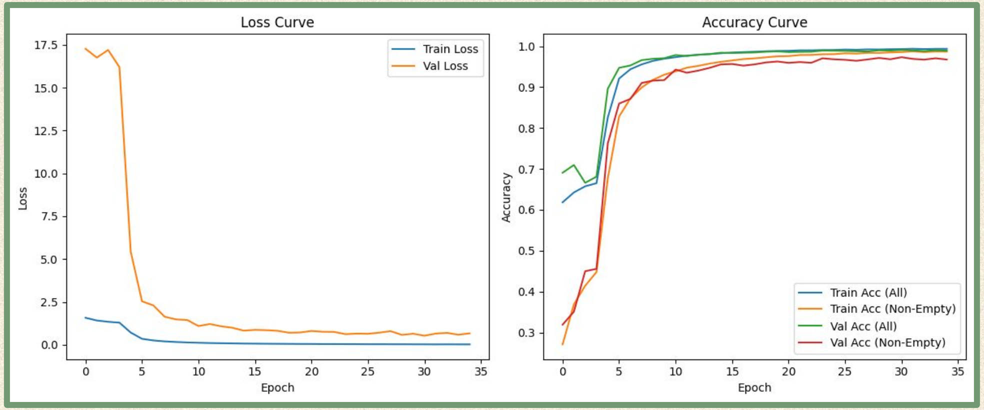
Our Model: 63.9%

Previous State of the Art: 15.6%

QUANTITATIVE RESULTS

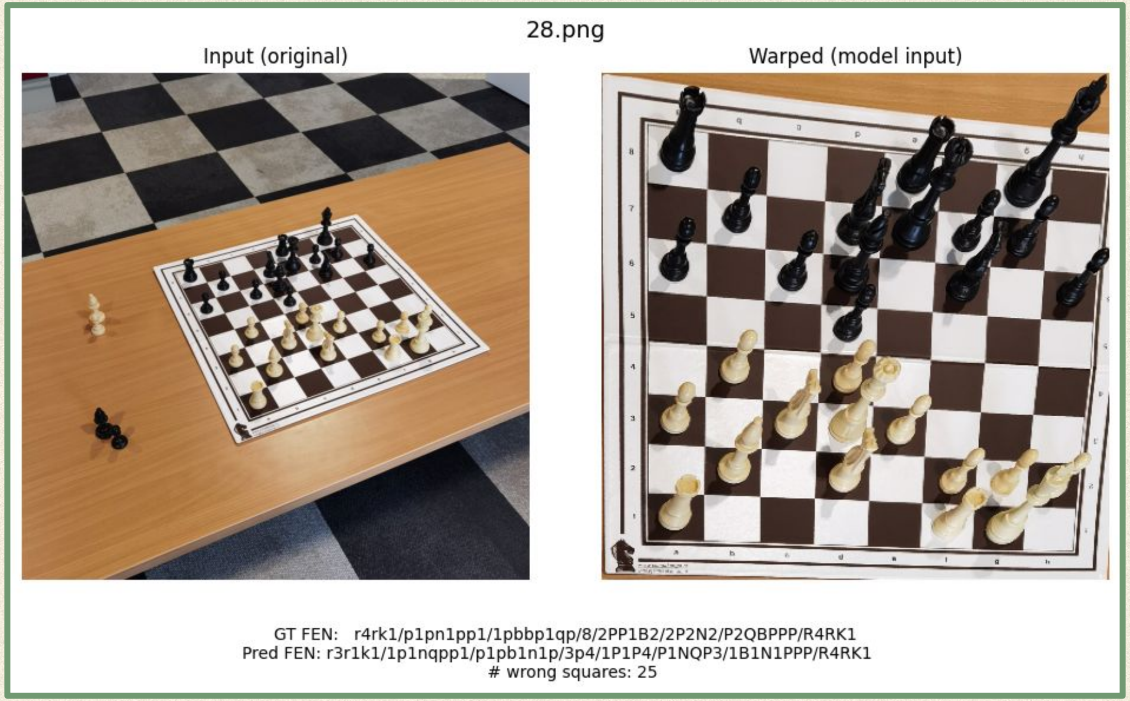


Confusion Matrix of Predicted vs. Actual Classes



Accuracy and Loss Curves during Training and Validation

QUALITATIVE RESULTS – INCORRECT



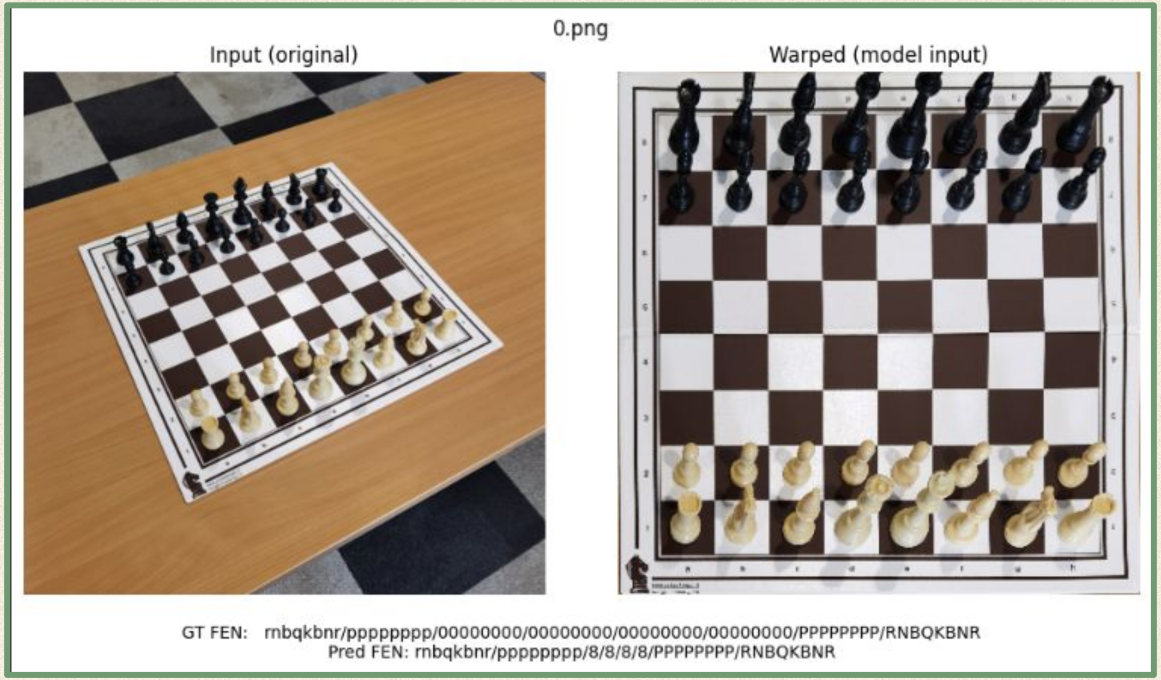
28.png

Input (original)

Warped (model input)

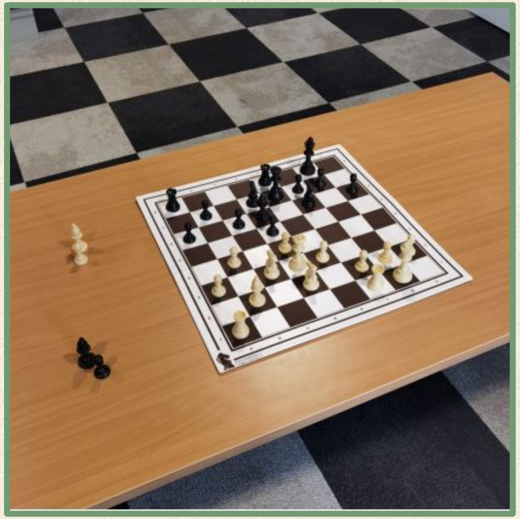
```
GT FEN:   r4rk1/p1pn1pp1/1pbbp1qp/8/2PP1B2/2P2N2/P2QBPPP/R4RK1
Pred FEN: r3r1k1/1p1nqpp1/p1pb1n1p/3p4/1P1P4/P1NQP3/1B1N1PPP/R4RK1
# wrong squares: 25
```

QUALITATIVE RESULTS – CORRECT

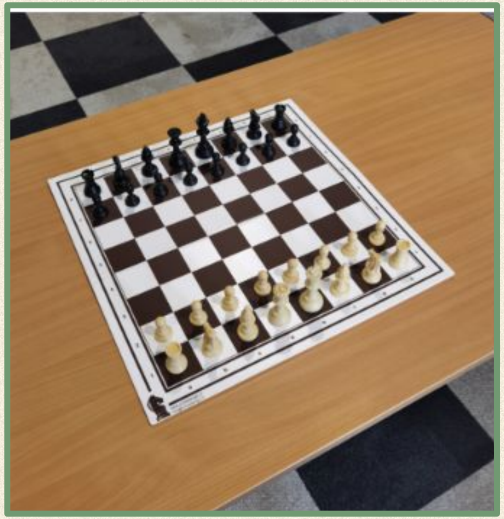


```
GT FEN:   rnbqkbnr/pppppppp/00000000/00000000/00000000/00000000/PPPPPPPP/RNBQKBNR
Pred FEN: rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR
```

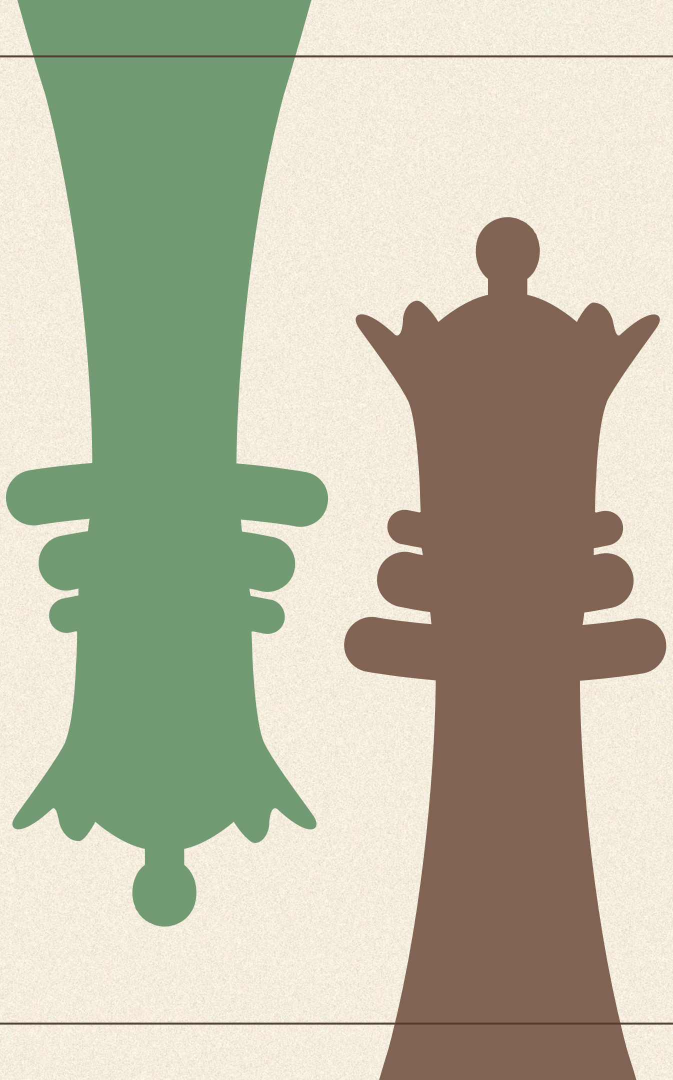
QUALITATIVE RESULTS – COMPARISON



Incorrect



Correct



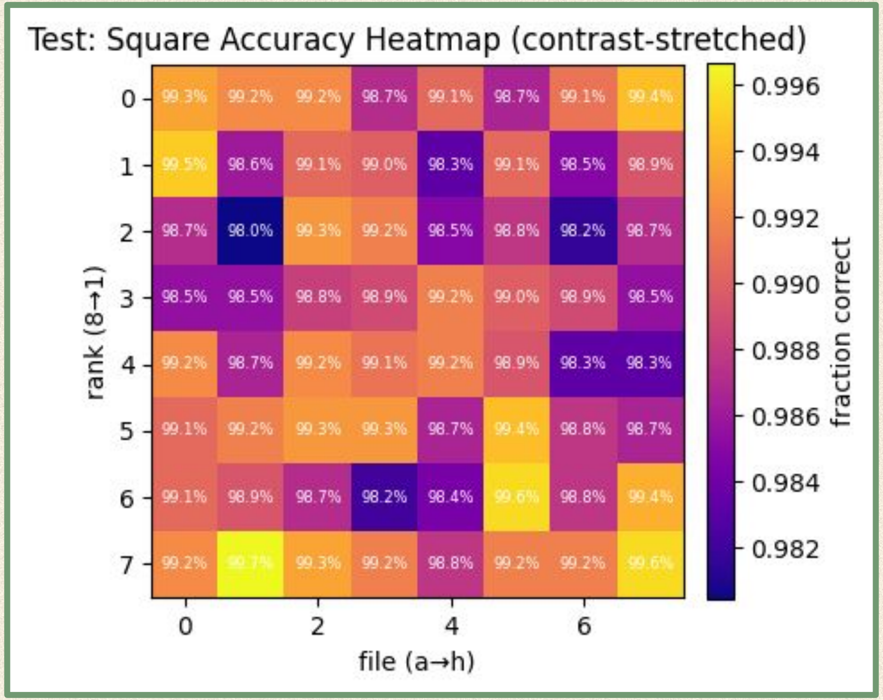
05

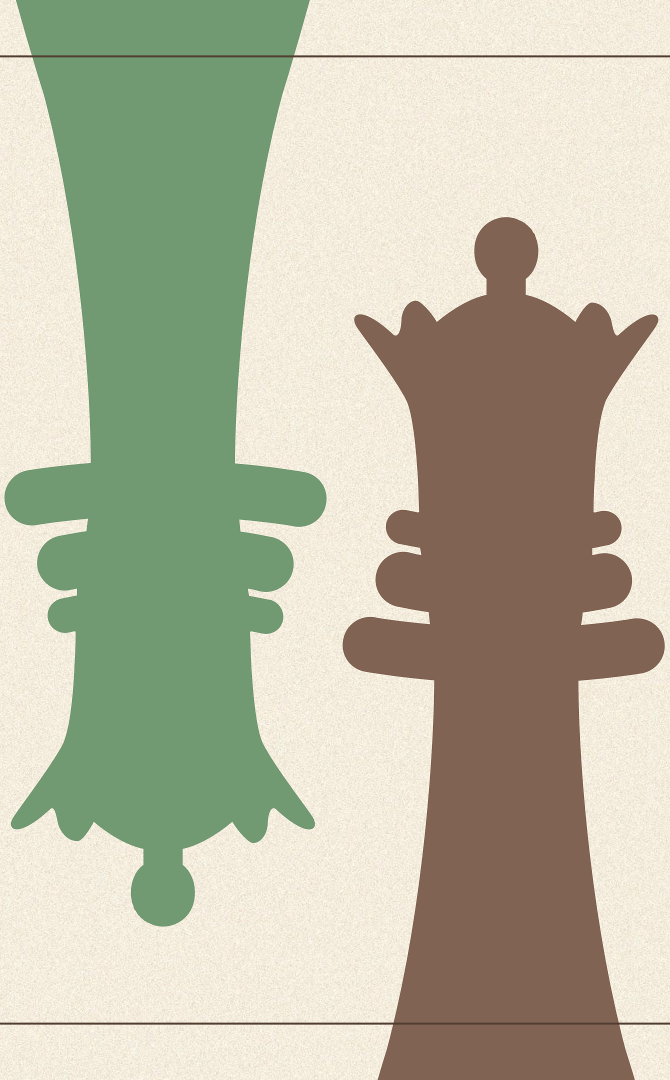
DISCUSSION

◆KEY INSIGHTS AND PERFORMANCE OVERVIEW ◆

- **Major performance gains** by **expanding training data** + stronger preprocessing pipeline
- **Without preprocessing** → **model fails** to generalize
- **~⅙ of images lost** due to failed board detection
- Takeaway → **Early pipeline planning** & dataset inspection are <u>critical</u>

◆HOW LOCATION OF PIECE AFFECTS ACCURACY ◆





06

DEMONSTRATION

REFERENCES