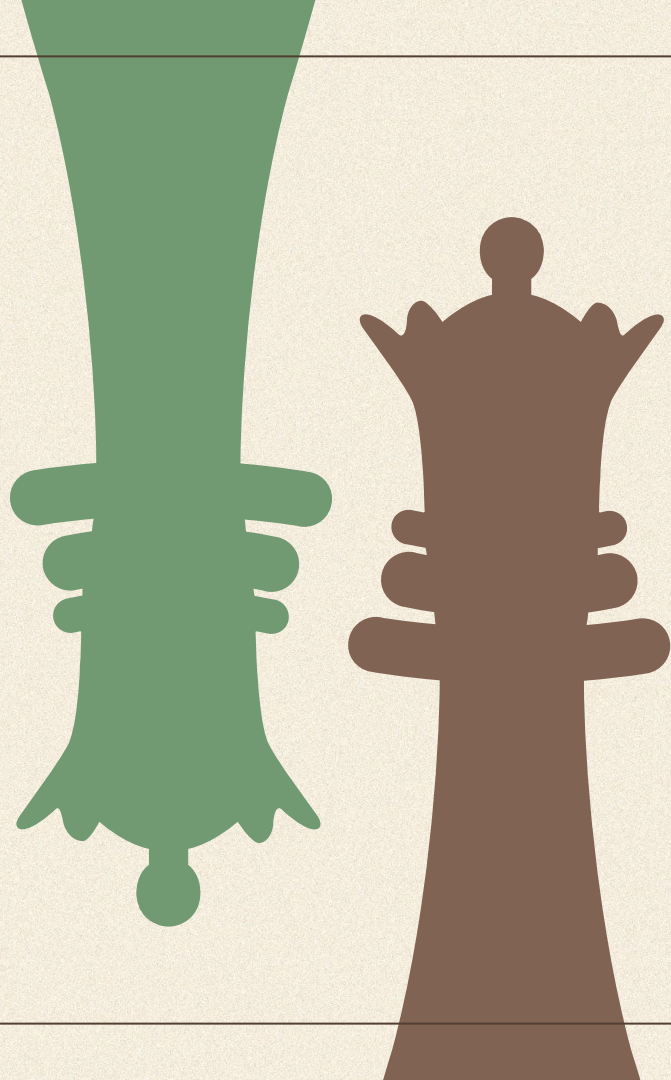




CVChess

Ved Patel, Gawthaman Senthilvelan
Luthira Abeykoon, Darshan Kasundra



◆ 01 ◆

PROBLEM

THE CHESS RENAISSANCE

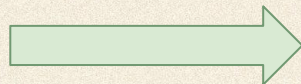


- **55% Increase** in Chess Tournament **Viewership** since 2020
- **Manually** keeping record of the moves of the game is **exhausting**
- **CVChess** proposes a way to **automate** the process

THE IDEA BEHIND CVCHESS



Chess Recognition Dataset (ChessReD)



r2qkb1r/pppn1ppp/4pn2/3p1b2/1PP
5/P2PN2/1B1P1PPP/RN1QKB1R

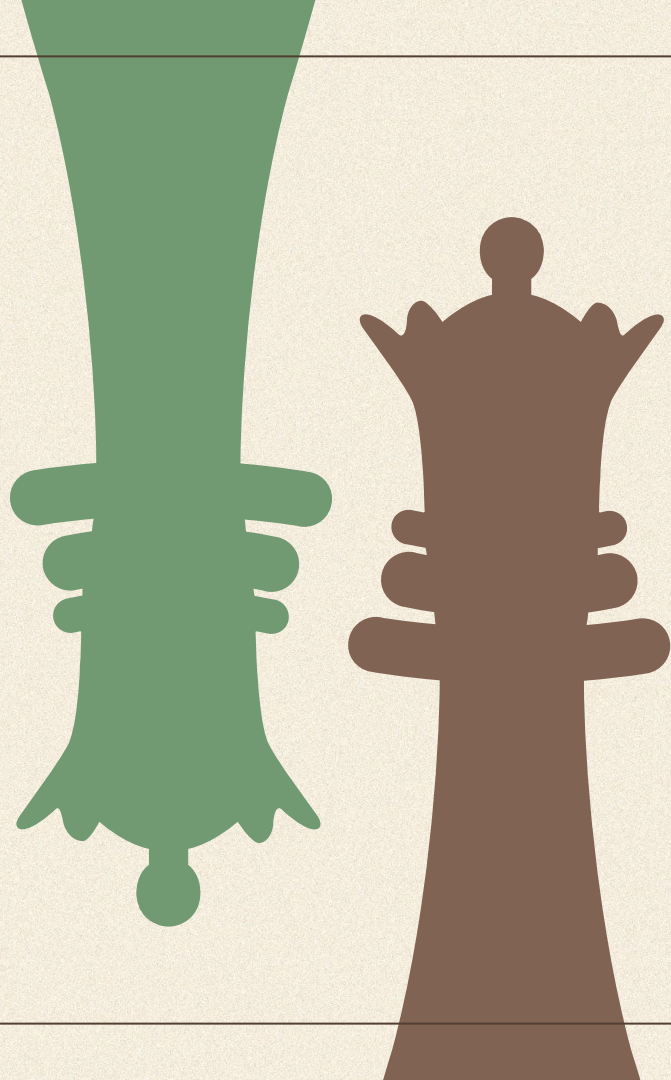
FEN Notation

◆ THE CURRENT STATE OF THE PROBLEM ◆

Metric	Blender Dataset		ChessReD		ChessReD*	
	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 ≤ 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 $\approx 7x$ 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

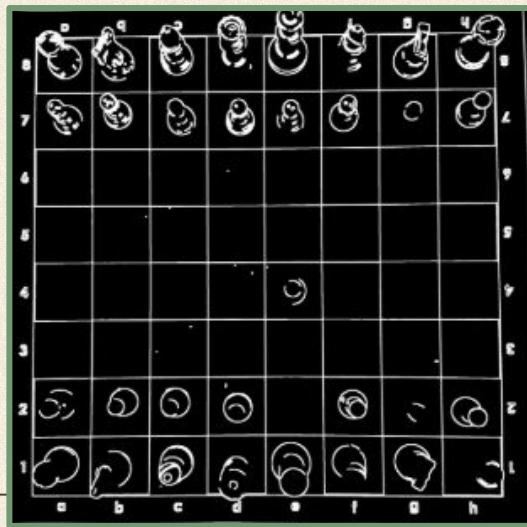
- Convert to **grayscale** → **Gaussian blur** (5×5) → **Canny edges** (50–150) → **Dilate**
- Extract **largest 4-corner contour** covering >5% of image
- **Arrange corners** so a8 is top-left, h1 is bottom-right
- Apply **cv2.getPerspectiveTransform()** to warp board to a 400×400 **top-down view**

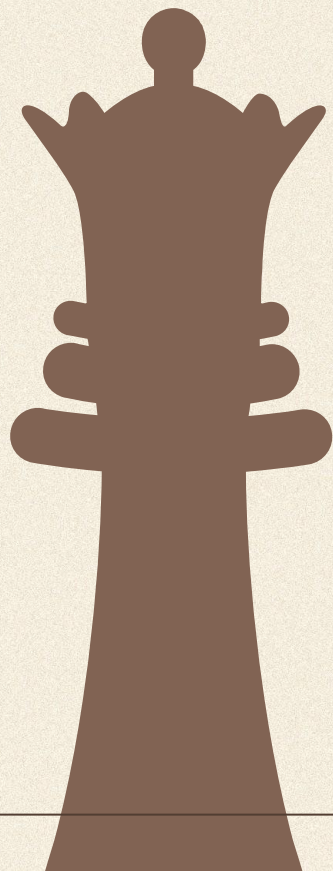
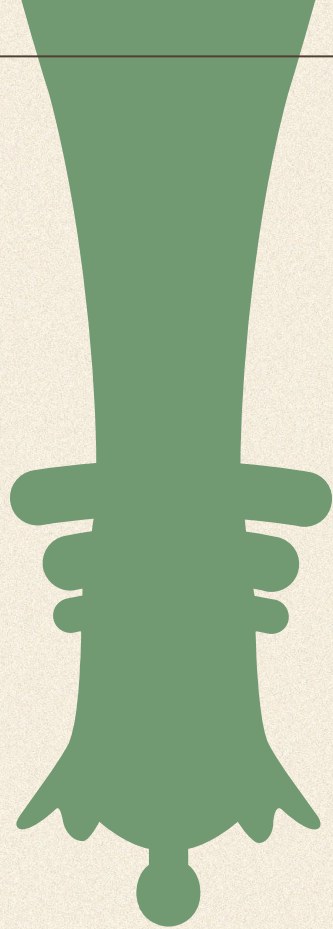
BOARD DETECTION AND PERSPECTIVE TRANSFORM



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 ◆

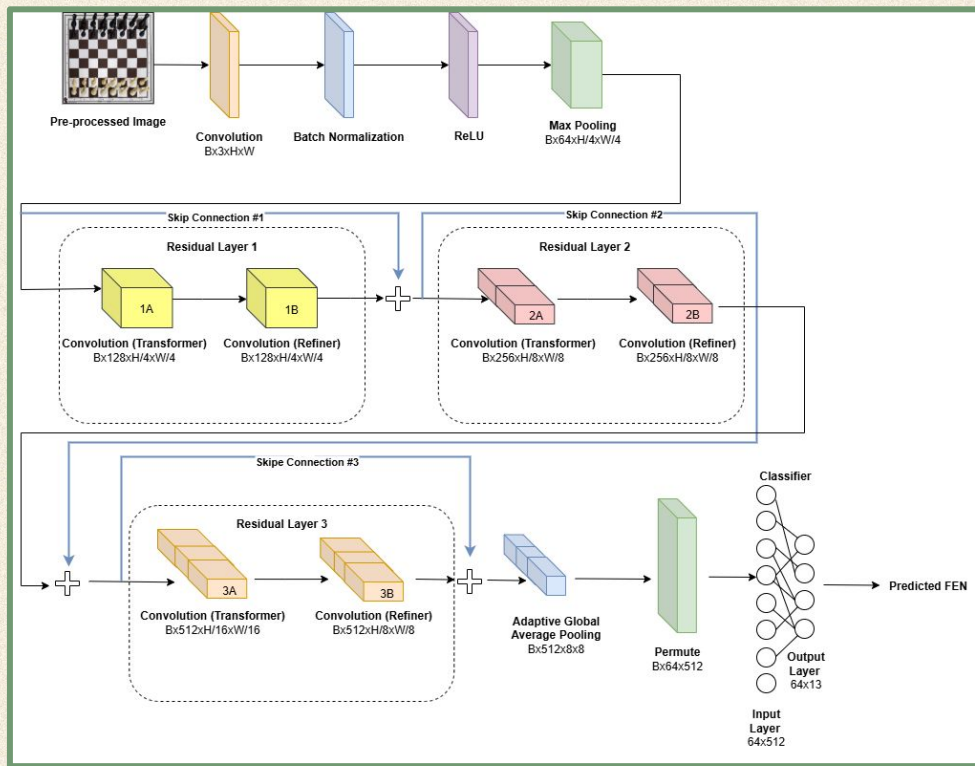
MODEL



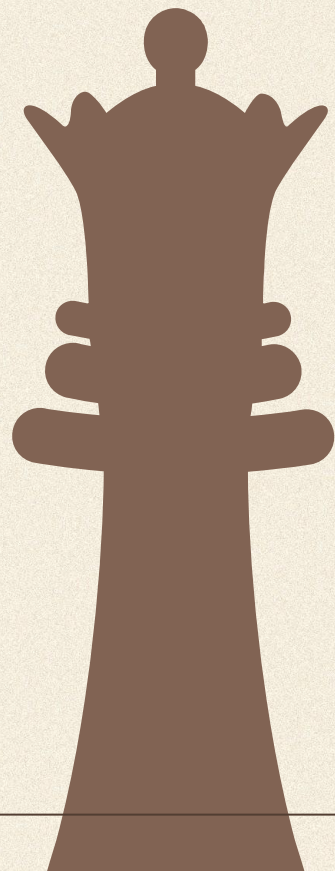
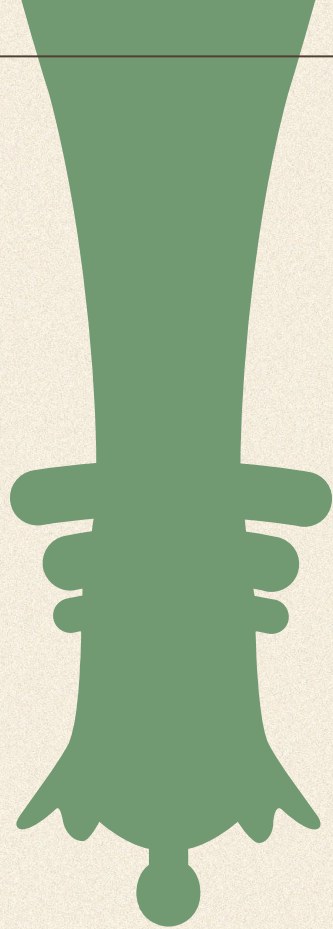
CVCHESS CNN ARCHITECTURE

- **CNN** architecture for image-based classification
- 256x256 warped board image → 64x13 (each square, 13 possible classes)
- Total # of parameters: 11,032,525
- Adam Optimizer (weight_decay=1e-5), lr=1e-4, batch_size=16, dropout=0.3

CVCHESSE MODEL ARCHITECTURE



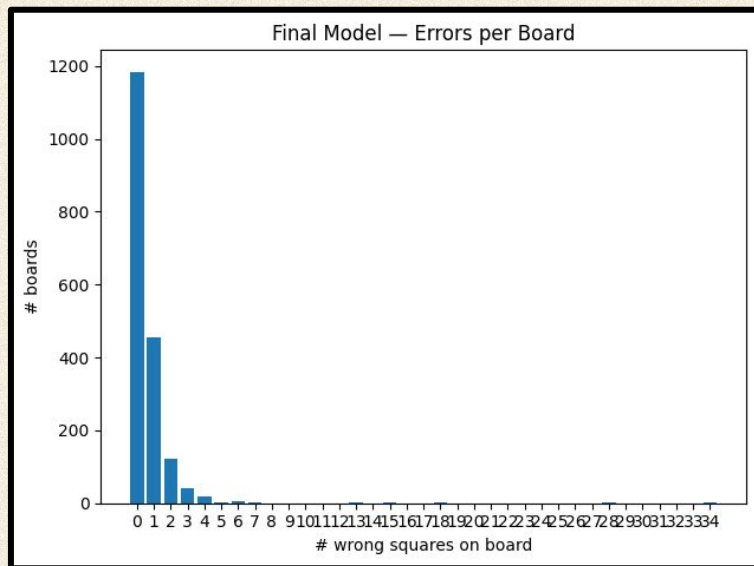
- Initial **Stem Layer**: 7×7 convolution + batch norm + ReLU + max pooling
- **3 Residual Blocks** for stable training, better gradient flow
- **Adaptive average pooling** to map directly to 8×8 chessboard grid



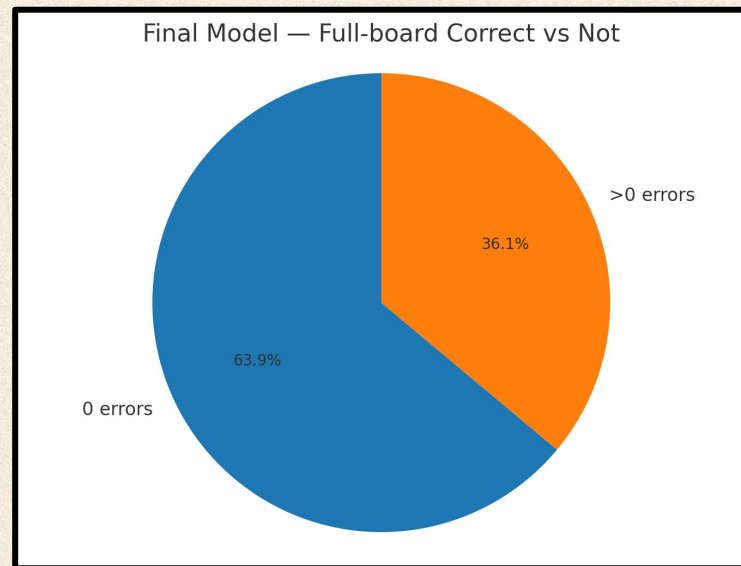
◆ 04 ◆

RESULTS

QUANTITATIVE RESULTS



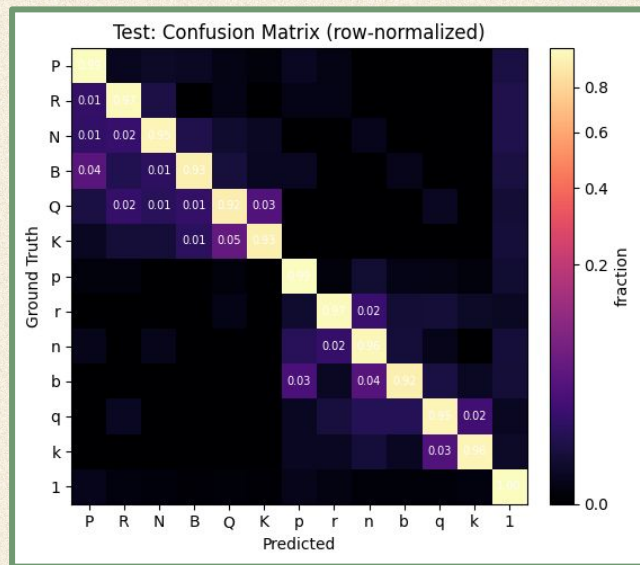
Frequency of Errors Per Board



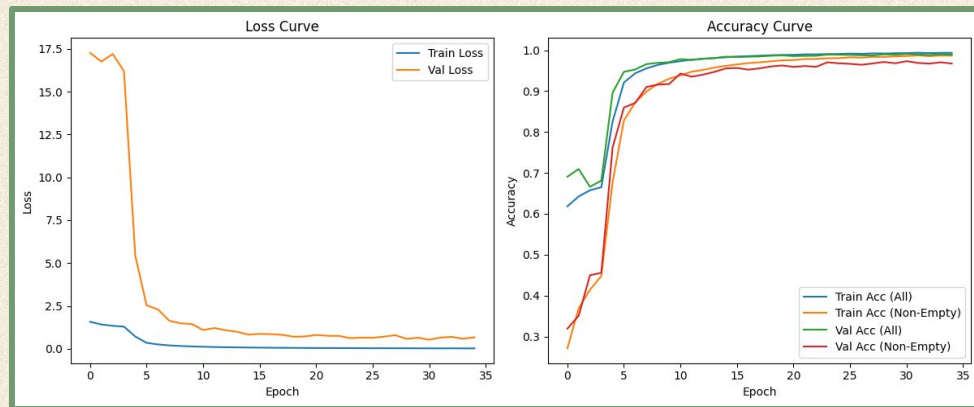
Final Model Accuracy

Our Model: 63.9%
Current State of the Art: 15.6%

QUANTITATIVE RESULTS

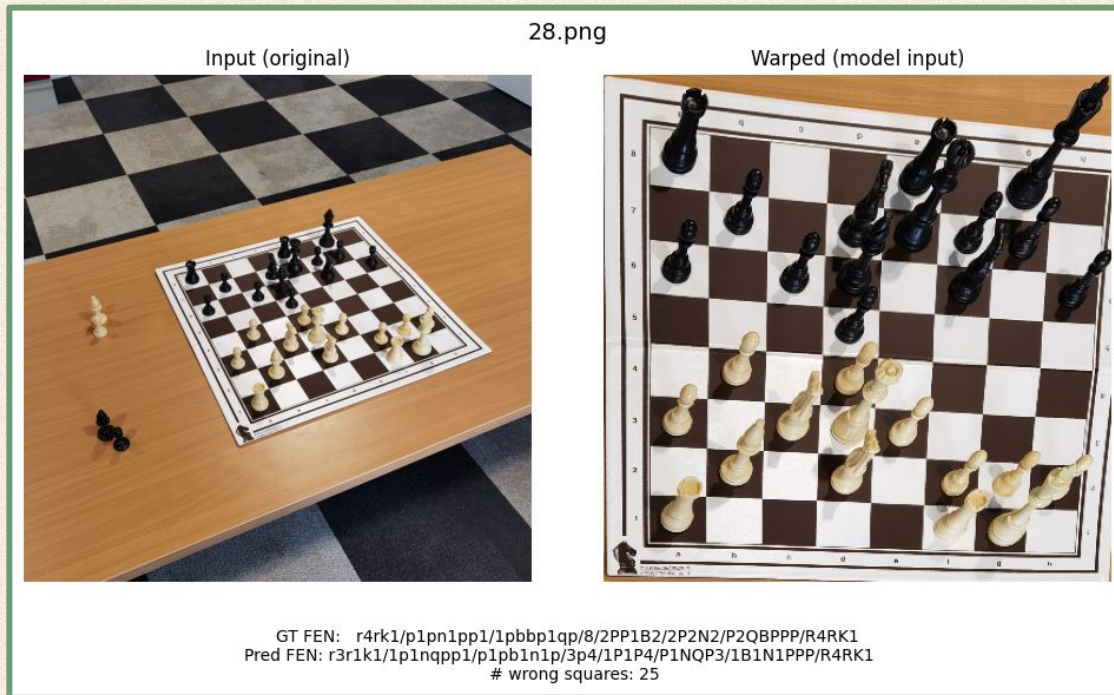


Confusion Matrix of Predicted
vs. Actual Classes

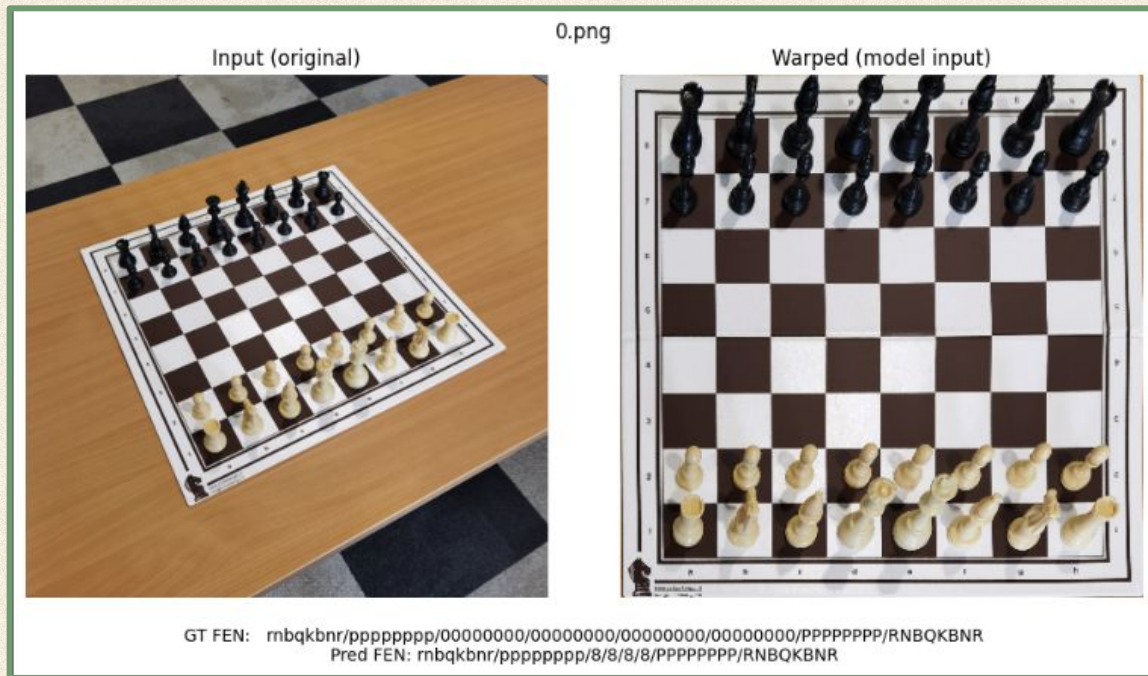


Accuracy and Loss Curves during
Training and Validation

QUALITATIVE RESULTS – INCORRECT



QUALITATIVE RESULTS - CORRECT



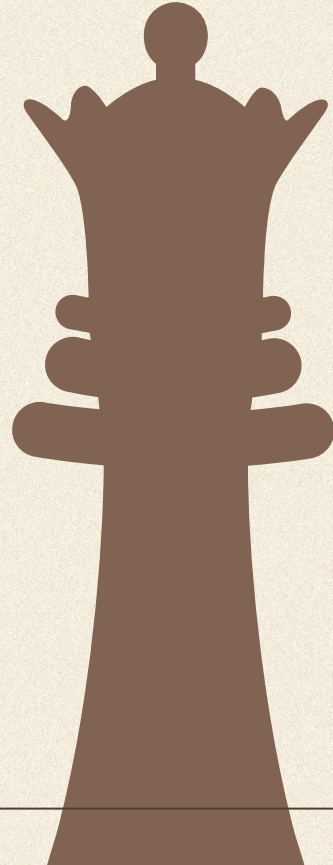
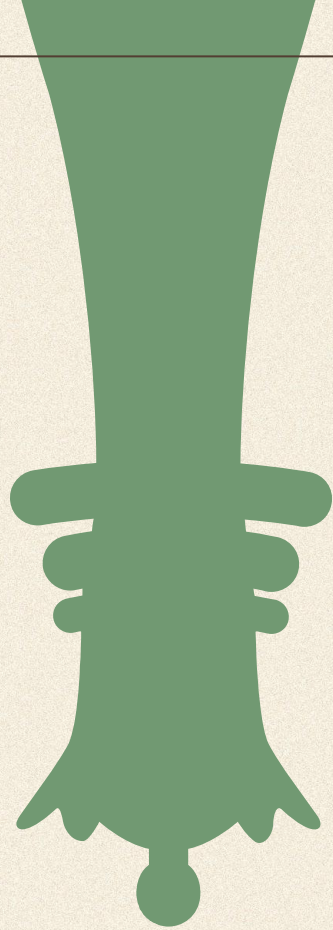
QUALITATIVE RESULTS – COMPARISON



Incorrect



Correct



◆ 05 ◆

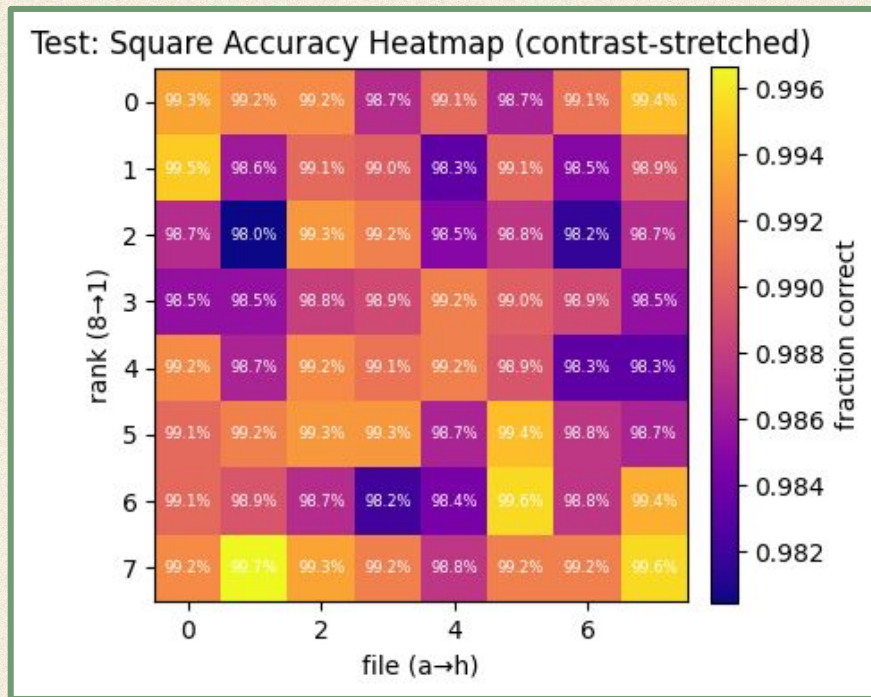
DISCUSSION

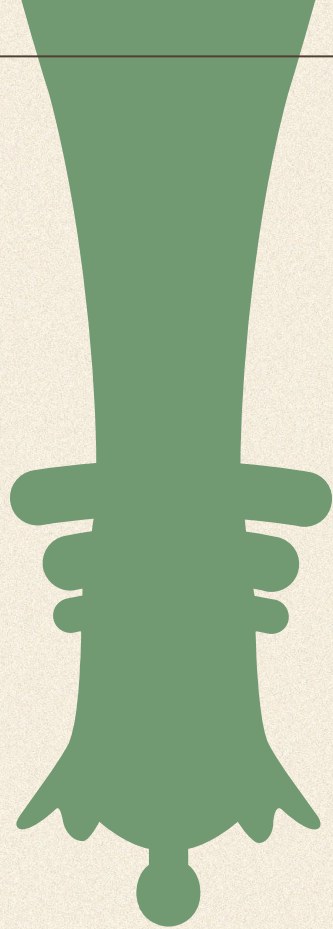


◆ KEY INSIGHTS AND PERFORMANCE OVERVIEW ◆

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

◆HOW LOCATION OF PIECE AFFECTS ACCURACY◆





◆ 06 ◆

DEMONSTRATIO

N

A decorative border composed of four chess pieces (two kings and two pawns) in dark green and brown, connected by thin lines, framing the slide content.

REFERENCES