



Abstract

- **Context:** Chess players take photographs of board positions for future analysis.
- **Purpose:** Our model turns photo into text. Player can paste text into a chess engine



Input Board image

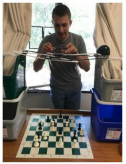
FEN Transcription:
2R5/1P2K1PP/8/4PP2/3N1p2/Pp1p1p/p5p1/1k5r

Model:

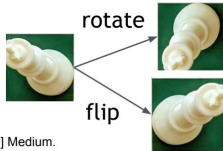
- Crop the image into squares of one piece
- Apply transfer learning from ResNet50
- Train a neural network on 15,000+ images
- Achieved average $F_1 = 0.94$

Data

- 10,000 labelled chess-piece images compiled by Daylen Yang [1].
 - Mostly pawns, empty
- We hand-labelled additional 10,000 images produced from the cropping algorithm in Model Task 1, then augmented as well



Data Augmentation



[1] Yang, D. (2016). *Building Chess ID*. [online] Medium.

Model

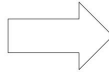
Task 1: Cropping



Input Board image



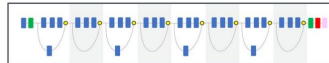
Transformed board



64 Cropped squares

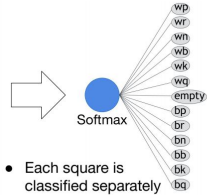
- Extract board from input image using Canny Edge detection, Hough line detection, corner extraction, then geometric transformation
- Evenly crop transformed board into 64 images of individual squares

Task 2: Piece Classification



Pretrained ResNet50

- Substituted last FC layer with 13 unit layer, softmax activation
- Adam optimization, Loss: sparse categorical cross-entropy



- Each square is classified separately and then the board is reconstructed



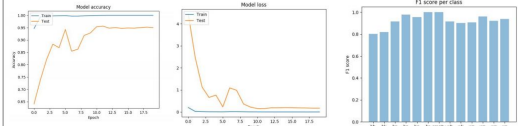
FEN Transcription:

mbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR

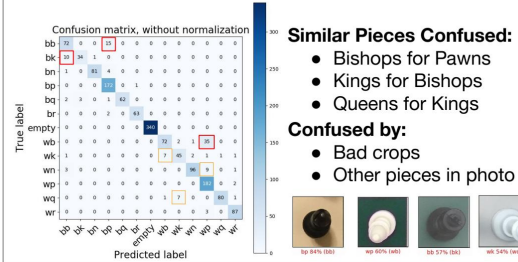
Results/Discussion

board mistakes	$F1_{train}$	$F1_{test}$	Trained Layers	Batch Size	Epochs	M_{train}	$M_{dev/test}$
≤ 2	0.934	0.944	7	256	20	~17323	~1295
≤ 4	0.955	0.954	14	256	30	~19789	~1398
≤ 15	0.983	0.984	33	128	20	~19789	~1398

- Best model achieves $F1=0.94$ with few mistakes on a real board transcription
- Other models overfit -> perform poorly on real boards



- Model inevitably overfits to limited data
- Best classification of empty squares and worst of black bishops and kings



Future Work

- Increase size of database (currently prohibitively small)
- Generalize model to more piece/board styles
- Build end-to-end object detector and classifier similar to YOLO

Acknowledgements

We would like to thank our TA Sarah Najmark for constant feedback and support. Also thank you to Professors Ng and Katanforoosh for teaching us.