



Efficient Chess Vision – Utilizing Predictive Weight Adjustments Based on Common and Logical Chess Positions



Allen Wu
Stanford University – CS231N

Introduction

- Chess has exploded in popularity over the pandemic
- There are over 500 million chess players worldwide
- Chess is very difficult to understand and learn
- Tools for analyzing and learning chess are very popular
- One such tool is to convert chess screenshots into chess FENs. This can also be used on videos to convert chess videos into FENs
- However, many existing chess vision AIs are computationally expensive

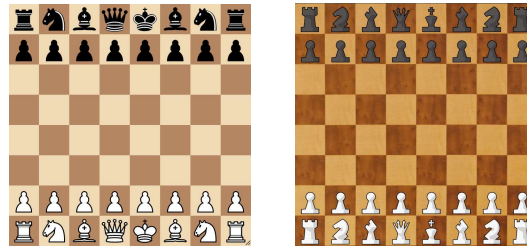
Problem Statement

How can a screenshot of a chess board be converted into a chess FEN depicting the same chess board using the least computationally expensive method available while still maintaining high accuracy.

Is it possible to improve a chess vision AI by adjusting weights based on what positions are most likely to appear in actual chess games?

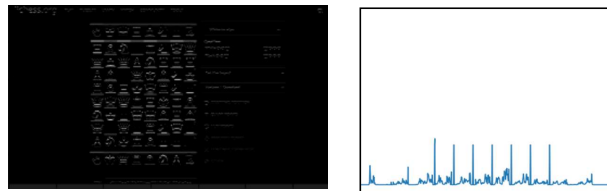
Dataset

The dataset includes screenshots from lichess.org using a variety of piece sets/board themes. Different backgrounds, zooms and compression settings were used



Methods

First, we use gradients in the x and y direction to find the board



Then, various classifiers were tried to classify the pieces on each of the 64 squares. Finally, manual weight adjustments were done on the last layer based on what is common in chess.

Experiments/Analysis

Without weight adjustments:

Model	Test acc
SVM	98.3%
SoftMax	97.94%
CNN – Small (10k features)	99.26%
CNN – Medium (40k features)	99.44%
CNN – Large (90k features)	99.14%
ChessvisionAI	99.64%

With weight adjustments:

Model	Test acc
SVM	98.32%
SoftMax	97.94%
CNN – Small (10k features)	99.24%
CNN – Medium (40k features)	99.5%
CNN – Large (90k features)	99.16%
ChessvisionAI	99.64%

Conclusions & Future Work

The model has comparable accuracy to the industry standard – showing that using a narrower training set allows for more specified applications

Weight adjustments can be beneficial, but further work should be done, as in this project, they were chosen manually, so they were not well optimized