Iowa State University
Electrical and Computer Engineering

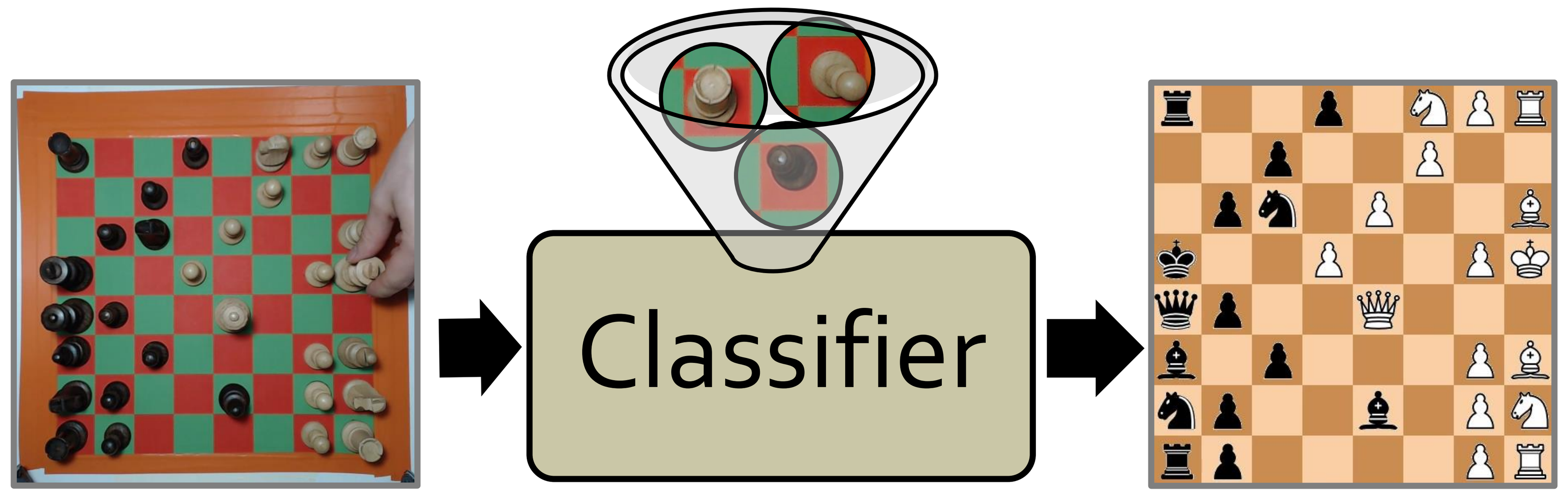
Honors Project
Spring 2018

Nathan De Graaf, Dr. Alexander Stoytchev (Advisor)

Computational Chess Piece Classification

Goals

- Build a classifier for a set of chess pieces.
- Use the classifier to parse and tabulate a chess game.



Background

This project started in Dr. Stoytchev's class on Computational Perception with the goal of tabulating a chess game. This was achieved by simply identifying which squares where occupied and tracking pieces from their starting position.

Adding a classifier removes constraints on the starting position and makes detection far more robust since frames are processed independently.

Isolating a Piece

We compare our frame with an image of an empty board. Pixels that are highly similar are ignored.

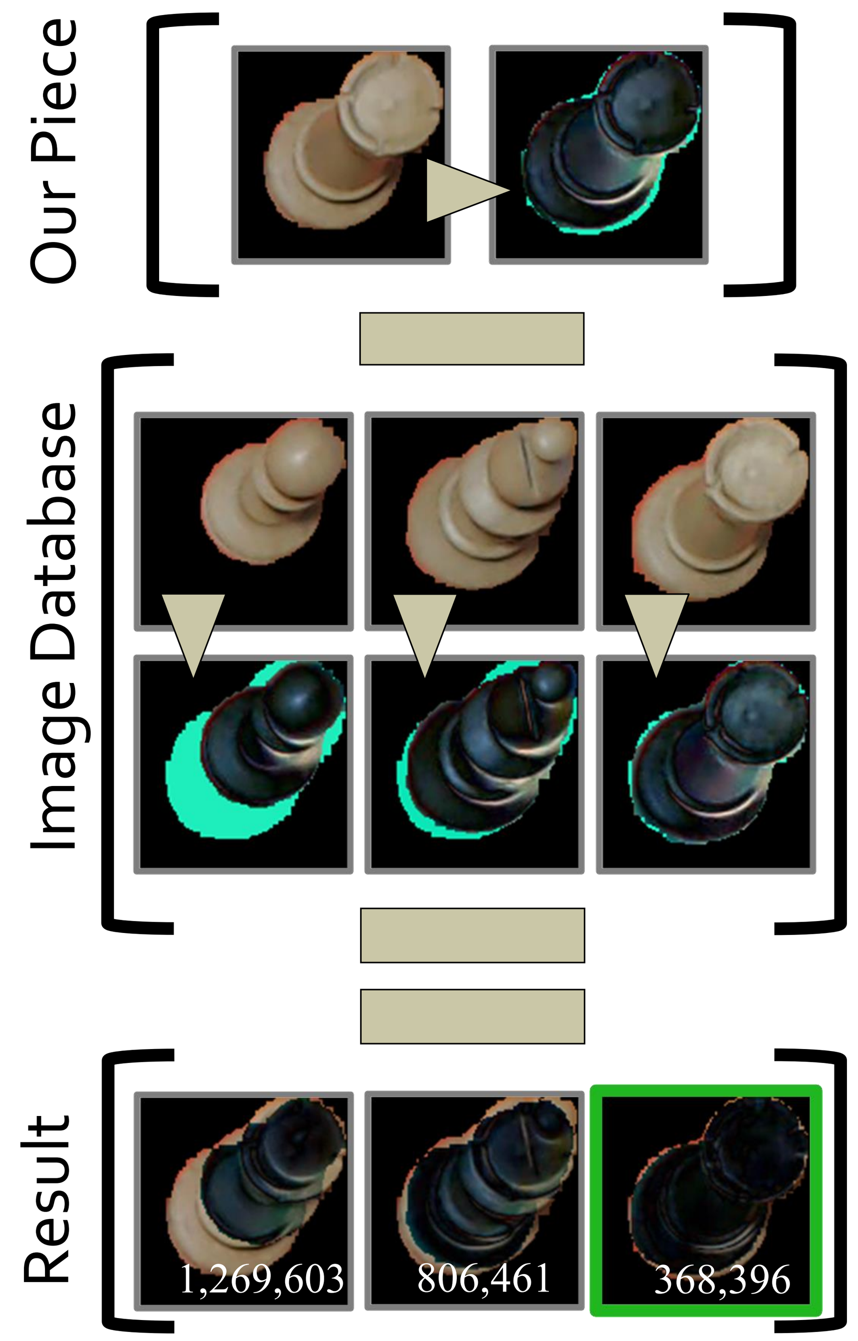
We remove pixels that:

- have a majority red or green component.
- match a predefined set of orange pixels.

For each board square, we crop out the associated region of the image in order to isolate its piece.

Finally, we clean up the piece by filling in gaps, removing unconnected pixels, and resizing the image.

Classifying a Piece



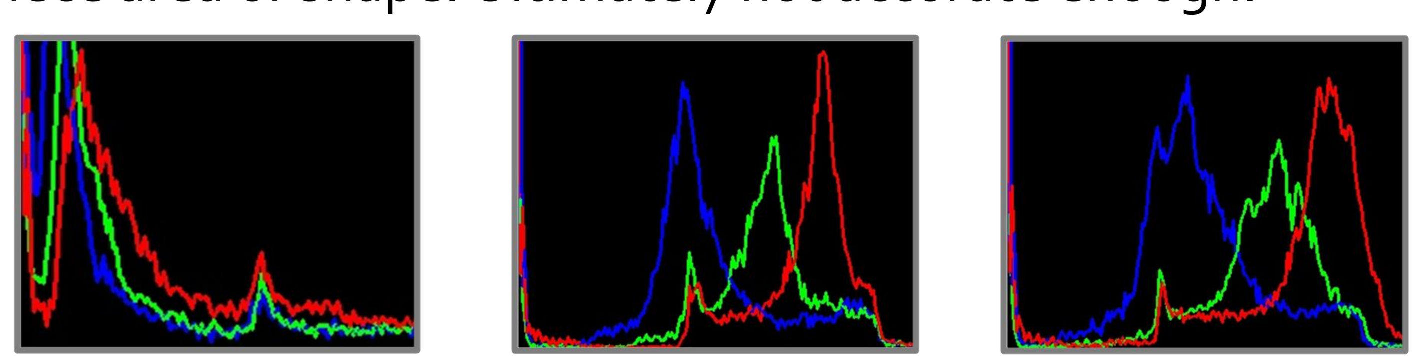
Consider the classification of a white rook in H1:

1. We load an image of every type of piece in this square (White pawn, bishop, and rook shown).
2. To correct for lighting, we represent images as deviations from their mean (dark, teal images).
3. Next, we take the difference between our unknown piece and the test images.
4. For each result, we compute distance as the sum of the pixels' brightness in the result.
5. The piece with minimum distance is our best match.

Alternative Classification Approaches

A number of alternate techniques were considered before differencing was chosen.

- Piece color histograms (pictured below). While successful for the white pieces, dark pieces did not have enough colors.
- Deep learning. Preconfigured models not specific enough, generally not configurable enough for this application.
- Piece area or shape. Ultimately not accurate enough.



Results

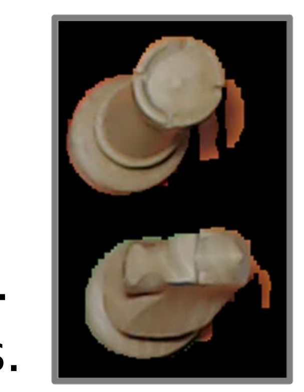
To create our final product, we combine our tools to isolate and classify pieces together with a complete database of images of every piece on each square. We teach our system to find the end of turns by watching the orange border.

Our system performs as follows:

- **84** correctly identified pieces from still images.
- **30** correct moves determined from videos: **800** correctly identified pieces from **3** minutes of video.

Limitations

A major limitation of our method of classification is that pieces must be almost perfectly aligned. Just centering almost works, except for cases such as the one to the left. We solve this by aligning images into corners so that the overlapping region extends to the center.



The second limitation of our system is its reliance on predefined colors. The image to the right caused poor identification due to incomplete orange color removal. We solve this by allowing easy color definition updates.

Conclusion and Acknowledgements

- Our system in its final form has achieved the original goals of this project with a very high degree of accuracy.
- There is room to improve how the system responds to deviations in lighting and alignment.
- Finally, I would like to close by thanking my advisor, Dr. Stoytchev, for kindly sharing his expertise and mentorship.