**FEN Chess Position Classification Using CNN**

|  |  |  |
| --- | --- | --- |
| Snir Lugassy  206312506 | Gal Oshri  311363873 | Eilon Amram  205679780 |

**Abstract**

Forsyth-Edwards Notation (FEN) is a common notation for chess board state representation. In FEN each piece is represented using a unique letter, The white pieces are denoted using upper-case letters and the black pieces using lower-case letters.

We will train a Convolutional Neural Network to classify the unique state of a Chess board in FEN notation given images taken from online Chess website.

**Forsyth-Edwards Notation Definition**

The full FEN will describe many details about a certain game, we will consider only notation used to describe the piece placement in the board.

The piece placement is defined from the white player perspective, using 6 groups of characters separated by a delimiter. We will use ‘-’ as a delimiter (adopted from the dataset) although the original definition suggests the usage of ‘/’ as a delimiter.

Each group of characters corresponds to a row in the board from top to bottom, each character corresponds to a piece or space in the row from left to right according to the following dictionary:

“p” = Pawn

“n” = Knight

“b” = Bishop

“r” = Rook

“q” = Queen

“k” = King

1-8 = Number of empty squares

White pieces are represented using upper-case letters and the black pieces using lower-case letters.

The state of the board in Figure 1 using FEN is:

**Chess Position Dataset**

Our dataset consists of 100,000 images of chess boards taken from the online website “Chess.com”, with a size of 400×400 pixels. The boards have variety of colors, and the pieces has a variety of both shape and color.

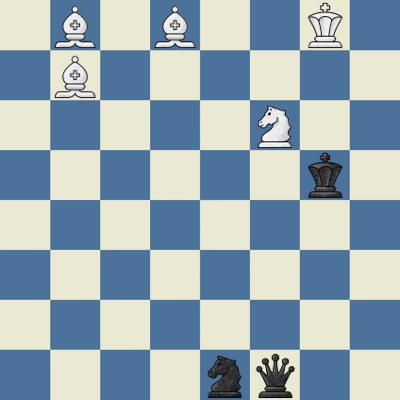


Figure 1: Example board image

The state of each board in FEN is stored in the file name of the image file.

**Data Processing and Transformations**

Since the Chess Positions Dataset was already cropped and labeled properly, the amount of processing over the data was minimal.

Image transformations

Each image file was converted into a tensor with the following shape:

The first and second dimensions represents a position on the chess board, the third dimension is the color channel of the image (RGB). The fourth and fifth dimensions are the color values of the image.

The color values were normalized into the range using .

Label transformations

The image labels, holding the FEN notation for each image, were converted into a label matrix holding values from the label set .

Each label corresponds to a possible piece on the chess board including an empty cell.

In order to convert FEN into matrix and matrix into FEN, we wrote the following helper functions:

|  |
| --- |
| fen2matrix(**board**):  pass |

|  |
| --- |
| matrix2fen(**board**):  pass |

**Model**

In our model the FEN classification task is split into smaller tasks, in which we classify the label of each cell using a CNN. The CNN’s input is 50x50 images of board cells, and the output a weight vector over the label set (vector in ).

In order to predict the label of the image we fed into the CNN we used LogSoftmax over the output vector in order to form a probability distribution over the label set.

The predicted label of the image which was fed into the CNN is

During training we have used LogSoftmax in order to calculate the probabilities.

Arch

**Inference**

using matrix2fen we output the FEN label of the board.

The above can be described using the following pseudo-code:

|  |
| --- |
| board2fen(**board**):  define **mat**[8][8];  for each row **i** in **board**:  for each cell **j** in **row**:  **mat**[i][j] = **CNN**(board[i][j])  return **matrix2fen**(mat) |

**Results**

- Accuracy results

- Hamming distance over the labels (dist.)

**Advanced Model**

- Random crop

- Normlization

- Dropout

**Discussion**

- Write about further improvements which can be done

- Write about testing legal boards missing feature

-