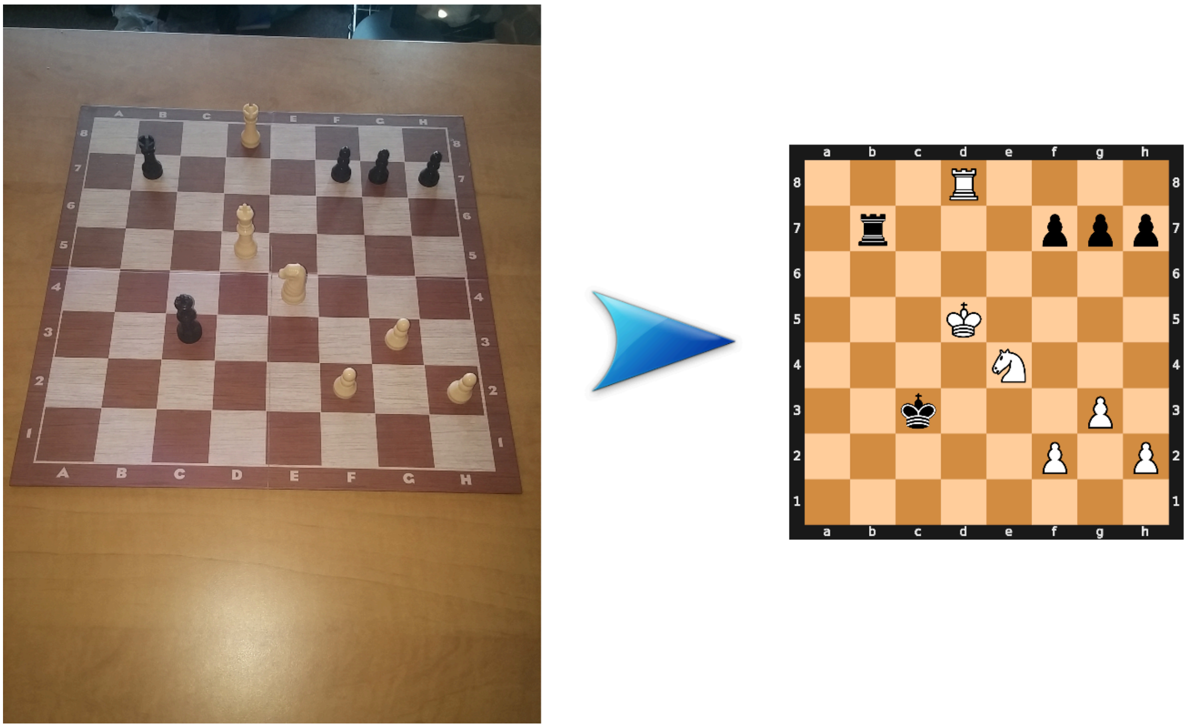
Fundamentals of Computer Vision (Undergrad) - B. Nasihatkon
Spring 1402 (2023)
K. N. Toosi University of Technology

# Detect the Chess Pieces Using Convolutional Neural Networks

Negin Mohammadi



## Overview

Your task is to write a program that detects the chess pieces in an image. This includes a rectangular bounding box around the pieces and the name of each detected piece. Further, you must detect the four corners of the board as the second task. Finally, you write a program that displays a graphical representation of the current board in a standard view in the third task.

## Dataset

You can download the dataset here.

## Task 1: Chess Pieces Detection

In this task, you need to do the following jobs:

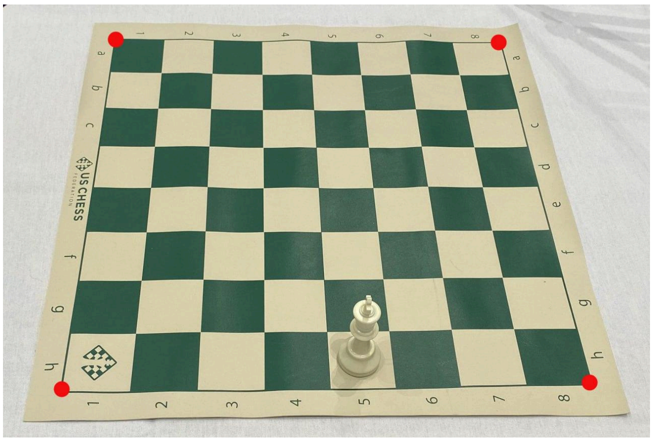
1. Read input data and preprocess to create training and testing data
2. Make an object detection CNN model to find chess pieces and their names.
3. Train your model using training data.
4. Test and display the history of the training results.

Notes:

- In the above dataset, the location of pieces and their name is labeled.
- You can use ready-made CNN models like YOLO, SSD, or retinNet. But note that although you have to train the model yourself and cannot use pre-trained models, you can use pre-trained weights as initial values.
- It is only allowed to use training data for the train. If you have hyperparameters, you can tune them using validation data but avoid using test data for the training process.
- You may use TensorFlow or Pytorch library and make your model using one of these tools. Although it is forbidden to use Keras, you can exploit Keras layers in TensorFlow.
- You are also not allowed to use compiled models and you should compile your customize model yourself.
- You will be asked about the architecture of the model and how to implement it during the presentation.

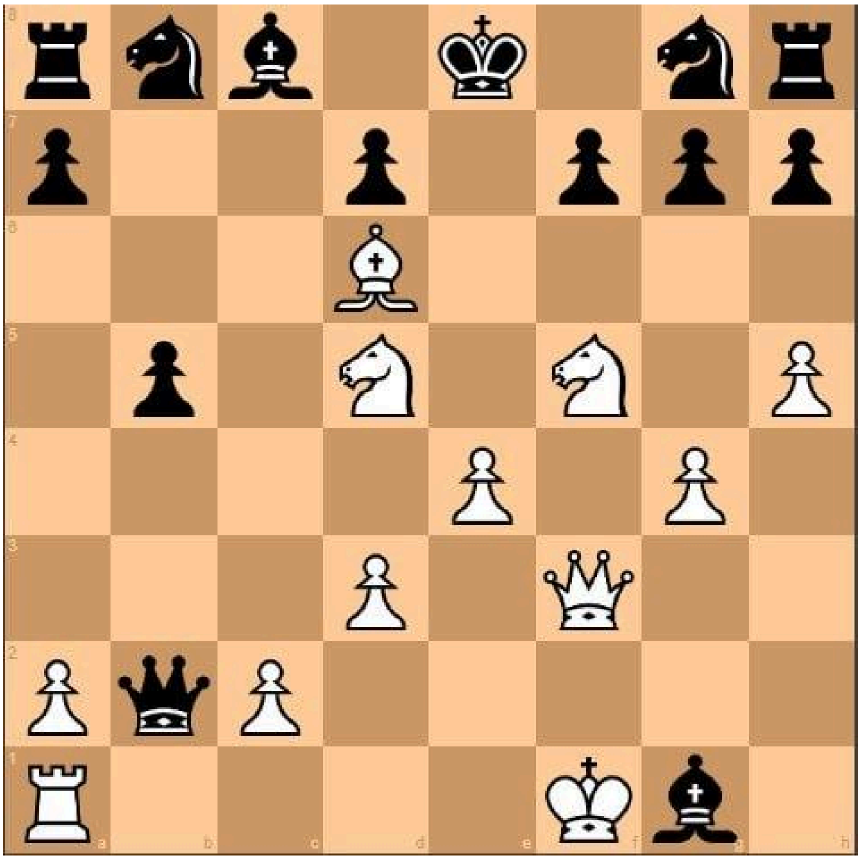
## Task 2: Chessboard Corners Detection

In this task, you need to find the four corners of the chessboard like the points marked in the photo below. For this goal, you should train your own simple CNN. There is no dataset for this purpose and you have to create a dataset by yourself. For example, you can write a program that saves the locations you click on the image. The number of outputs in this CNN should be 8, that is, the coordinates of 4 corners that can be raw or normalized.



## Task 3: Graphical Representation of Chess Pieces in a Standard Way

In this task, According to the corners that you obtained from task 2, you can find a perspective transformation that maps obtained corners to four corners of the standard view. Using that transformation you are able to find the location of each piece in the virtual chessboard and finally show the chessboard like the image below.



Notes:

- If you did not do task 2 or did not get accurate results from it you can manually declare the location of the four corners to complete task 3.

## Teamwork and Presentation

- You may work in teams of at most 2 students. However, each student will present the project to the TAs individually. Each student must be able to fully explain all parts of the code.
- Besides the other parameters, your program is evaluated by testing multiple images visually at the time you are presenting it to your TA.
- Your programs will be checked for similarity. Similar codes will not be marked.

## Useful Links

Neural Networks in Tensorflow

Implementing SSD in Keras