Go Board State Recognition

Team Project on the course “Introduction to Computer Vision” by
Bair Mikhailov and **Pavel Bartenev**

Skoltech

Problem

- Go is a chinese abstract strategy board game for two players in which the aim is to surround more territory than the opponent. The game is widely known, more than 46 million players know how to play it

- Streamings and recordings of offline matches during the tournaments require special boards or extra people to convert the position into digital format

- We propose our fully software solution, which everyone with any camera can afford

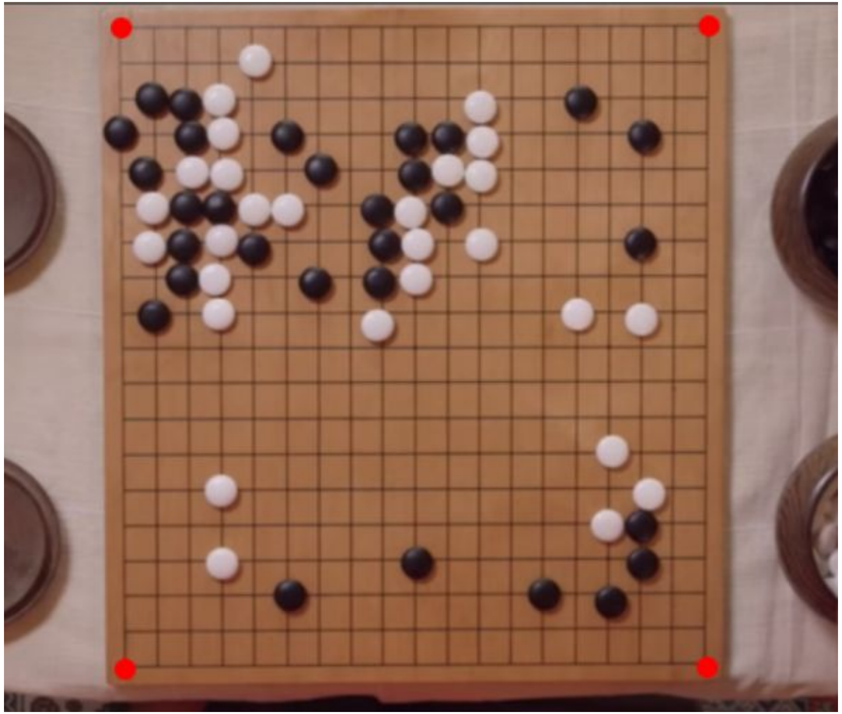
Datasets

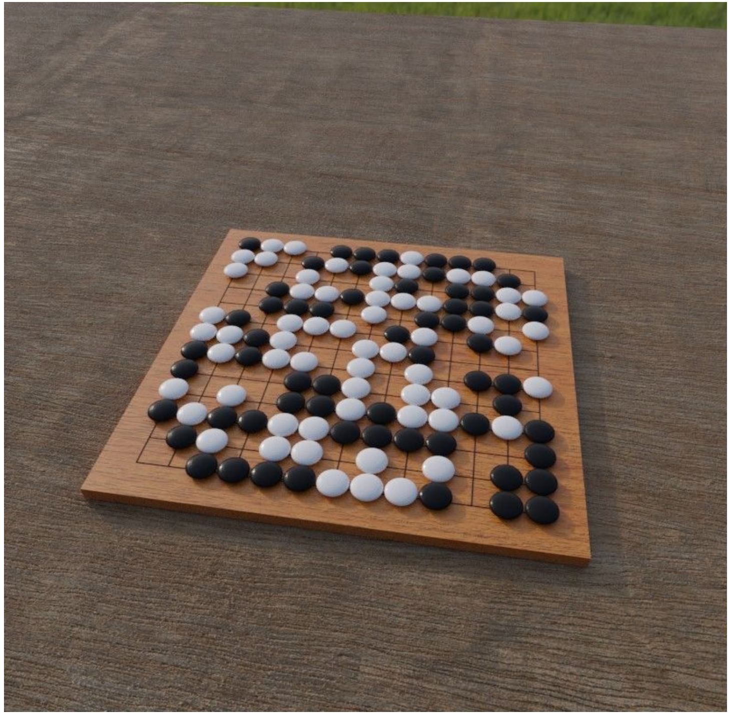
We used **2** datasets:

1. Gomrade dataset: go (baduk) images with annots by Davids
    a. 2000 real images, 19 x 19 boards
    b. Has fully annotated stone positions and exact board placement

2. VisualGo by Luan Ademi
    a. 2000 synthetic images made in Blender, 13 x 13 boards
    b. Has fully annotated stone positions

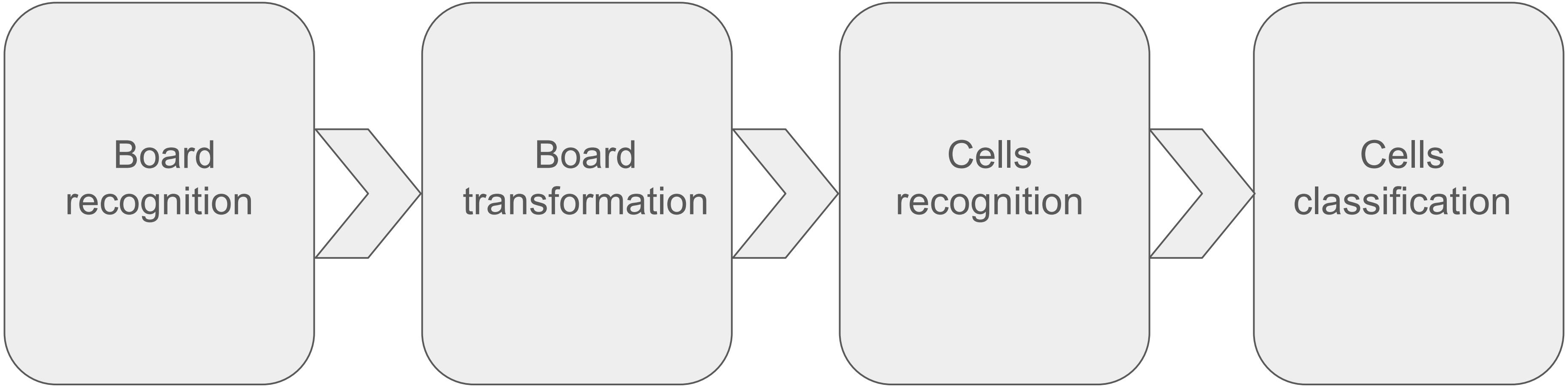
Examples





Skoltech

General pipeline



Board recognition

Board transformation

Cells recognition

Cells classification

Skoltech

Board Recognition

1. **DL approach**: Finetune YOLO model to find four corners of the go board.

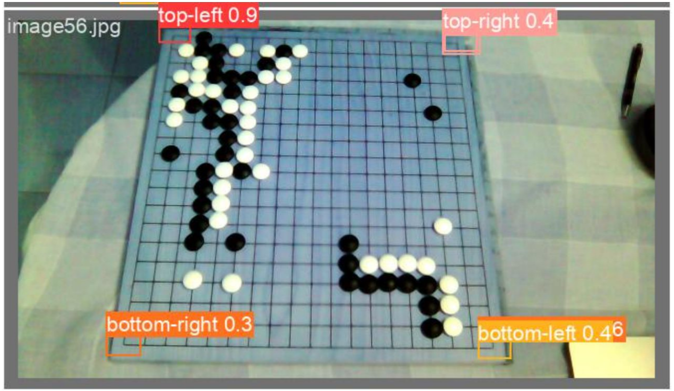
   - Pros: more robust to different types of the board, lightning and other conditions

   - Cons: slower, a large enough dataset is required

2. **Classical approach**: Canny edge detector + Hough transform + DBSCAN clusterization

   - Pros: faster, requires less data

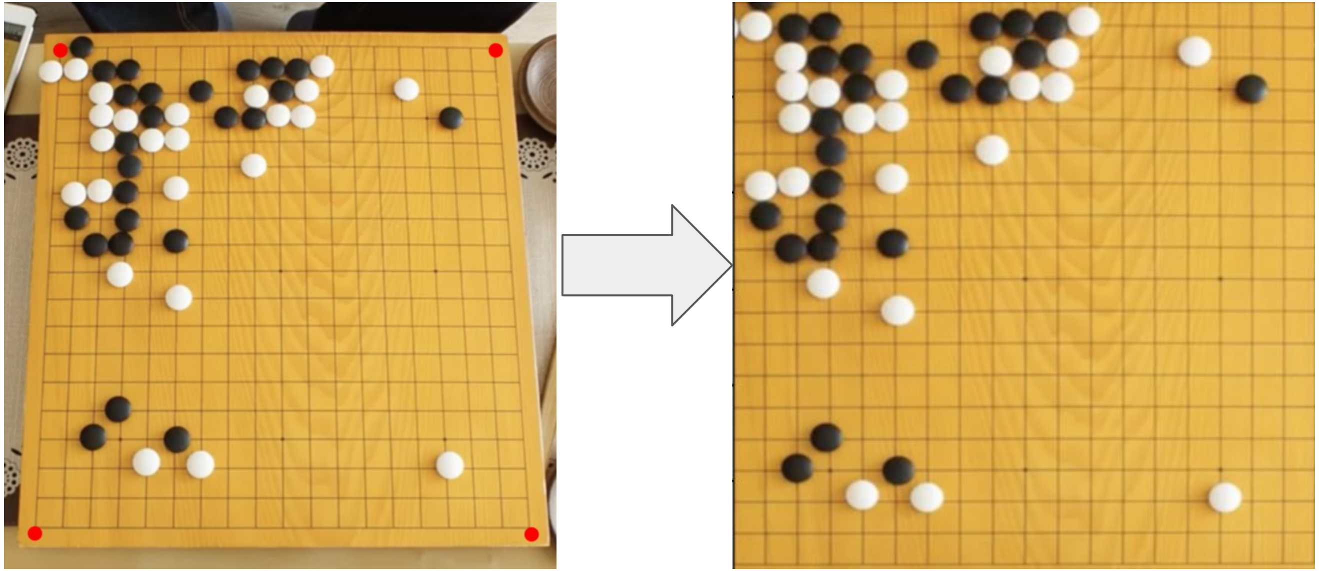
   - Cons: extremely hard to tune hyperparameters



Skoltech

Board Transformation

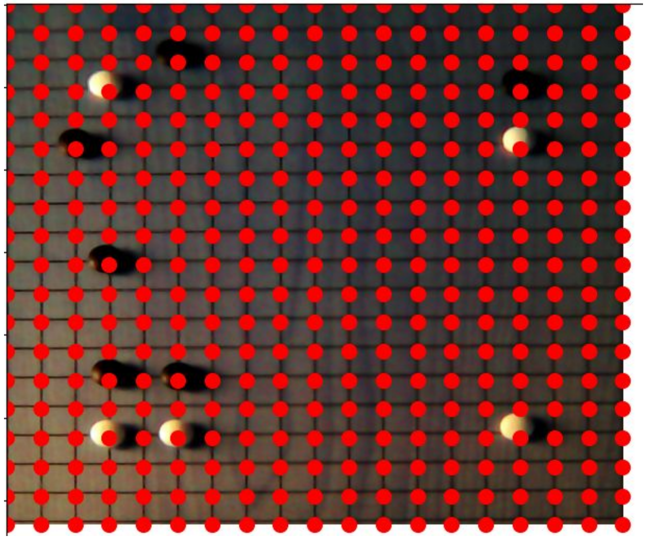
We transformed the board using a 4-points transformation to a rectangle



Cells Recognition

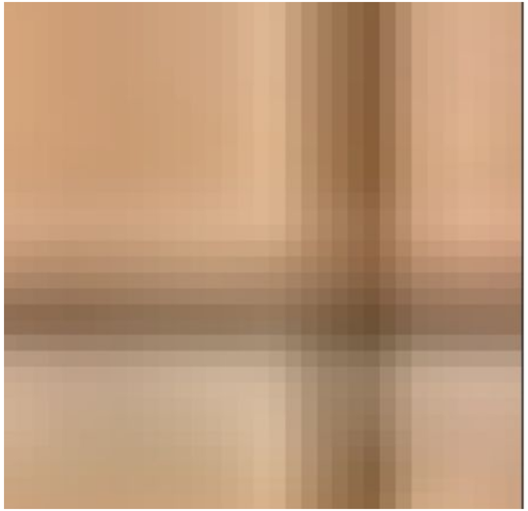
Uniform grid

We can employ the knowledge of the number of cells and their equal sizes to split the transformed board using uniform grids for each axis.



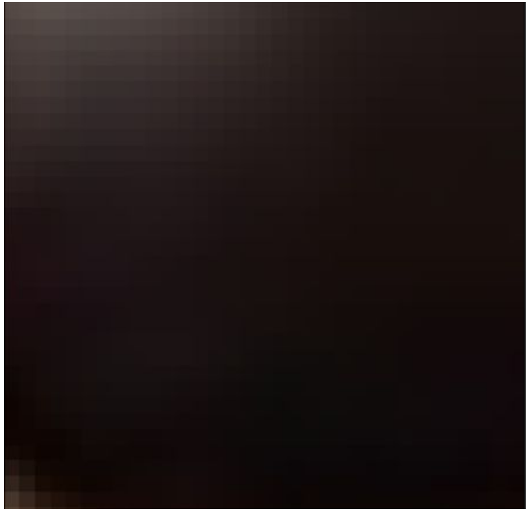
Skoltech

Cells classification



No stone

White stone



Black stone

Cells classification

We tested 2 potential variants:

1. **Threshold approach:**

   Calculation the median pixel in grayscale and use 2 thresholds to classify the cell

   - Pros: fast, no need to train (or easily trained)
   - Cons: lightning-dependent

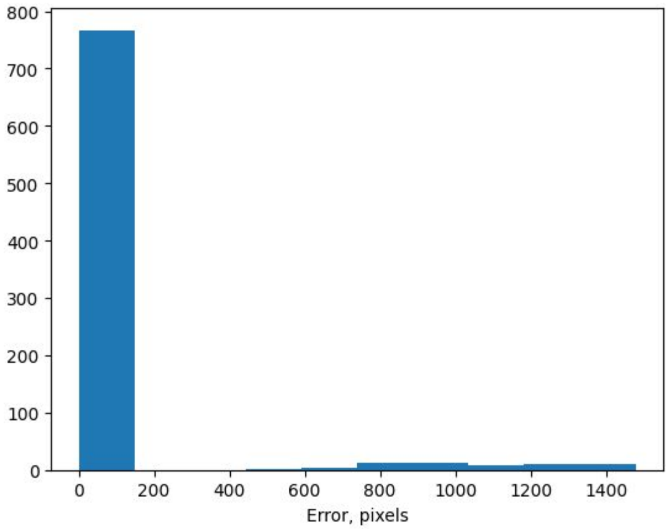
2. **DL approach:**

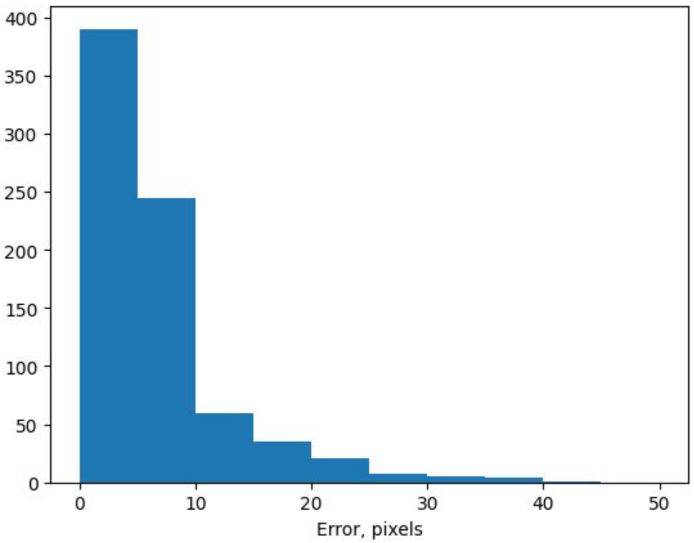
   ResNet-18 with 3 classes

   - Pros: better accuracy, less dependent on conditions
   - Cons: need of training

Final results

In **80%** of the cases the error in the estimation of corners position is less than **10 pixels**

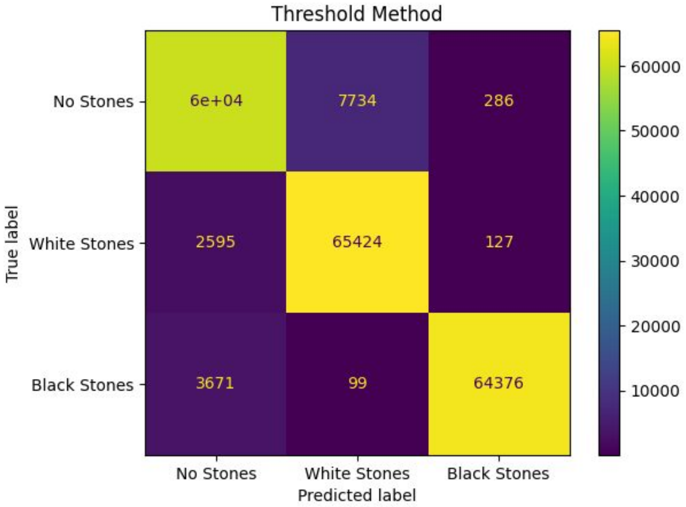


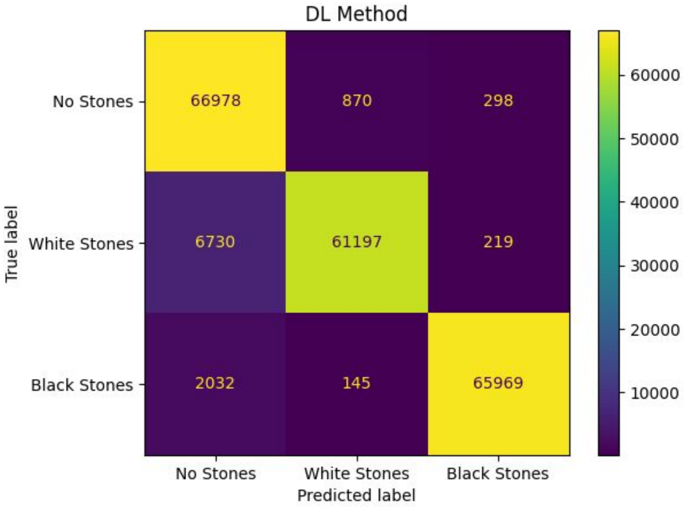


Skoltech

Final results

	Accuracy	F1-macro
Threshold approach	0,939	0,939
DL approach	0,949	0,949





Skoltech

Potential improvements

- Collect more diverse dataset of go boards on order to better train the DL models

- Change the cell recognition algorithm to be able to determine different board sizes

- Work on lightning: come up with the solution to normalize lightning conditions across image

Thank you for your attention!