CHESS DETECTOR AND ADVISER

NIKOLAY IVANOV, BOGDAN ALEXANDROV

Skoltech
Skolkovo Institute of Science and Technology

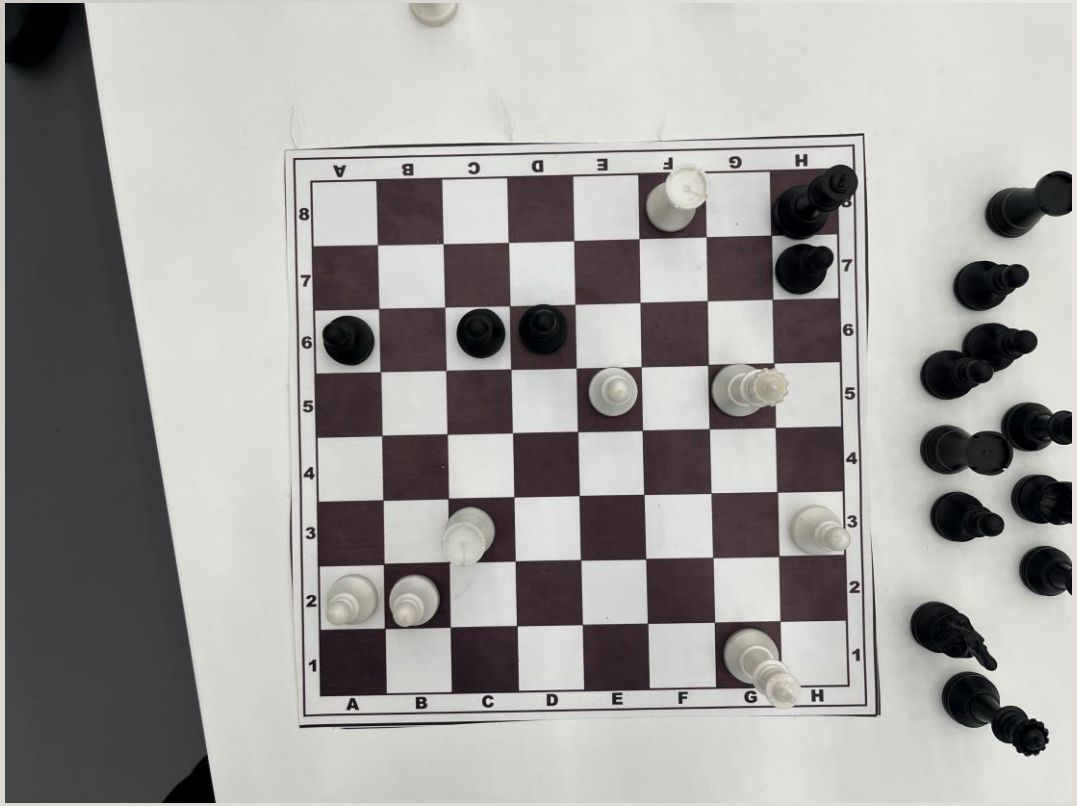
Problem Statement

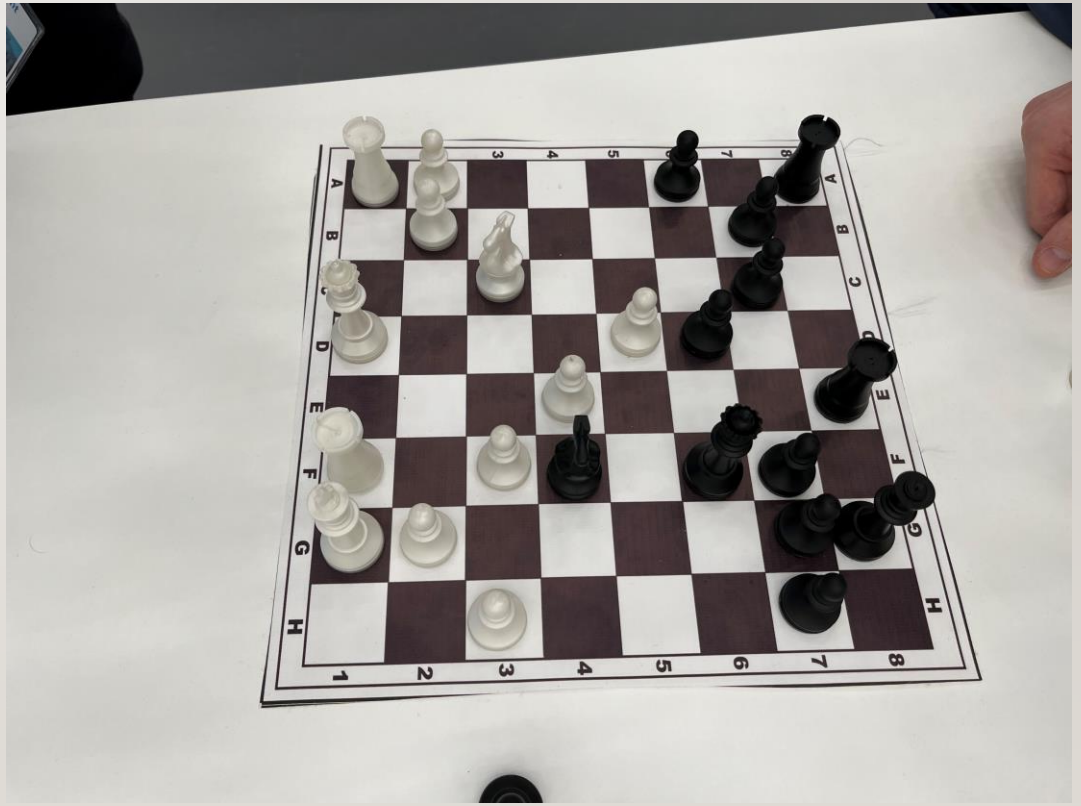
People who play chess really like to improve their skill of playing, and play on a real board instead of a computer. They can play against computer or rather get advises for a next step, also it's possible to store database of moves and analyze the game afterwards.

That's why we want to develop algorithm, which detects chess figures and predict future step for players, analyze the position etc.

Dataset

Photos of chessboard made from different angles with good quality, obtained manually from Skoltech library

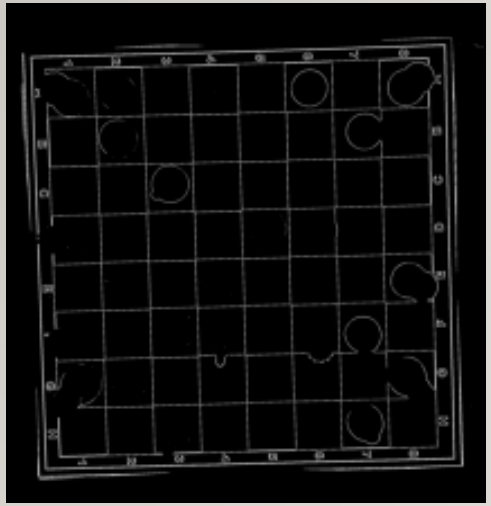




Pipeline of algorithm with motivation

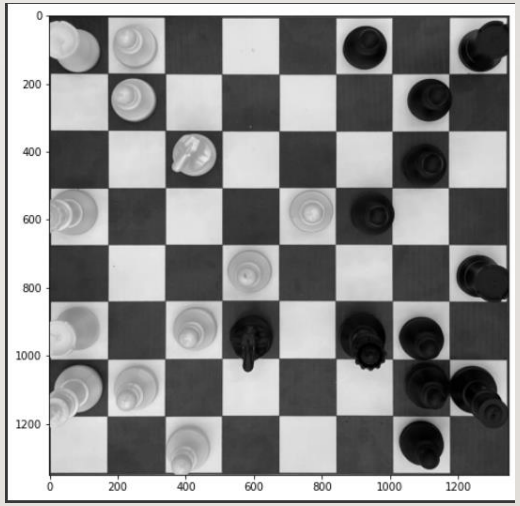
1) We need to detect chess on one concrete board, that's why we used:
Gaussian Blur + AdaptiveThreshold + FindContours + ApproximatePoly from OpenCV
2) To detect figures correctly we need good view of them, so we transform it with ProspectiveTransform
3) We need to obtain state of a game, so we cut the whole image by 64 equal cells,
Then used NN classifier (5 Conv Layers + linear) to classify our pieces
4) Used chess engine to advise best move and estimate position

Pipeline step by step

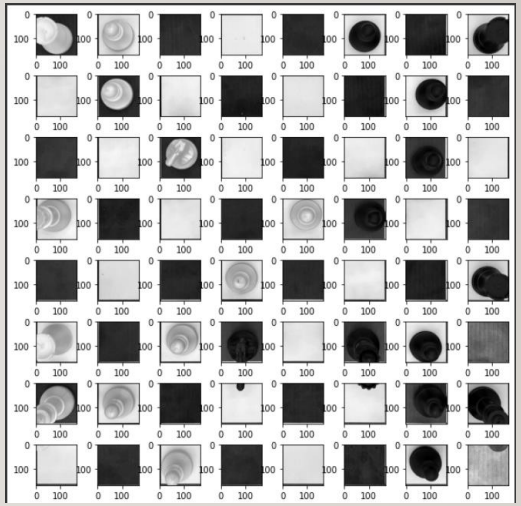


Gaussian Blur
+ AdaptiveThreshold

FindContours



ProspectiveTransform



Pieces for classification

Tools of our project

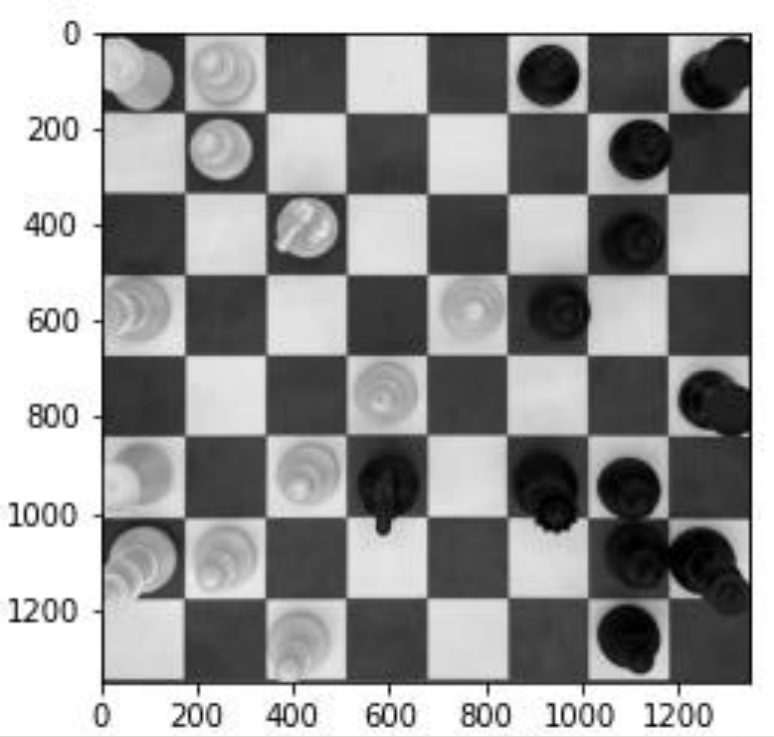
1) OpenCV https://opencv.org/
    1) Blurring
    2) Thresholding
    3) Finding Contours
    4) Prospective transformation

2) PyTorch https://pytorch.org/
    1) Creating DataLoader
    2) Creating Neural Network classifier
    3) Prediction

3) Chess engine https://python-chess.readthedocs.io/
    1) Analyze environment
    2) Find available steps
    3) Suggest one of them

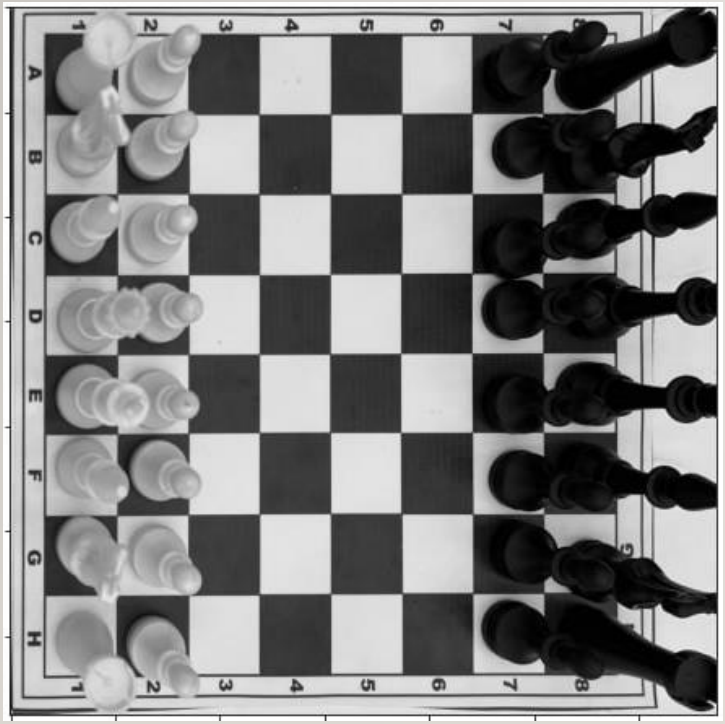
4) NumPy https://numpy.org/
    1) Arrays using
    2) Plotting with matplotlib

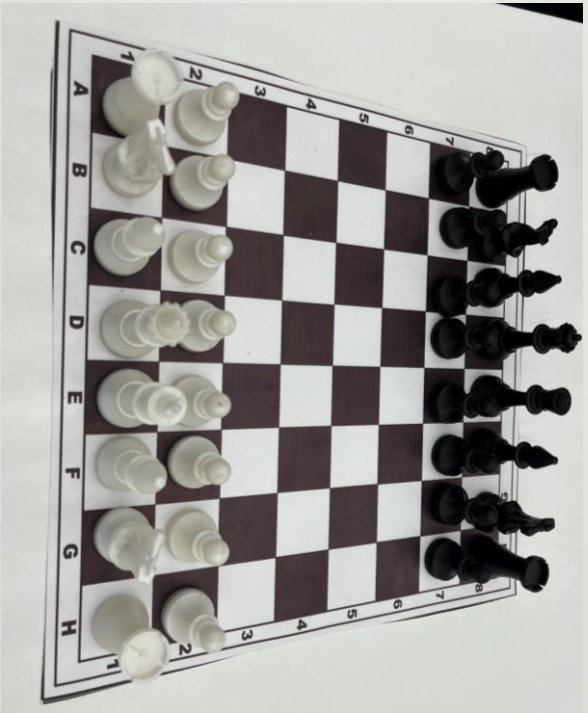
Good examples



Our classifier works very well for nice clear images from the top of the board, the shapes don't have to be right in the squares.

Bad examples

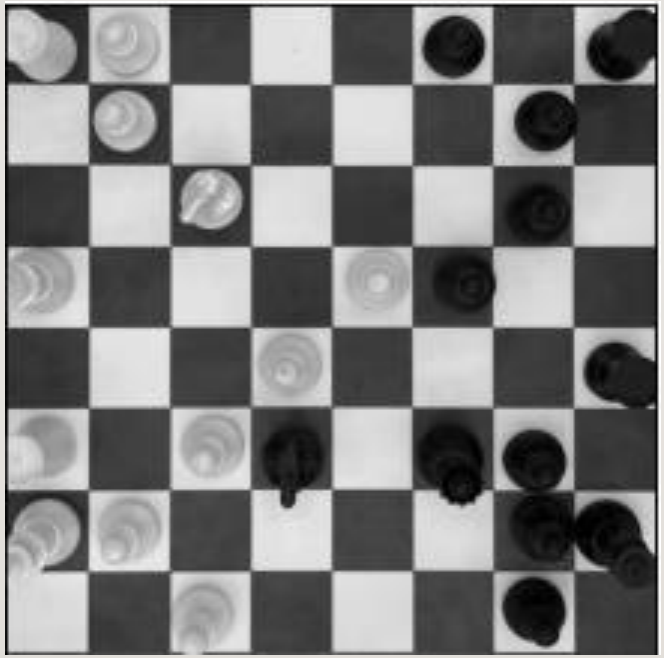




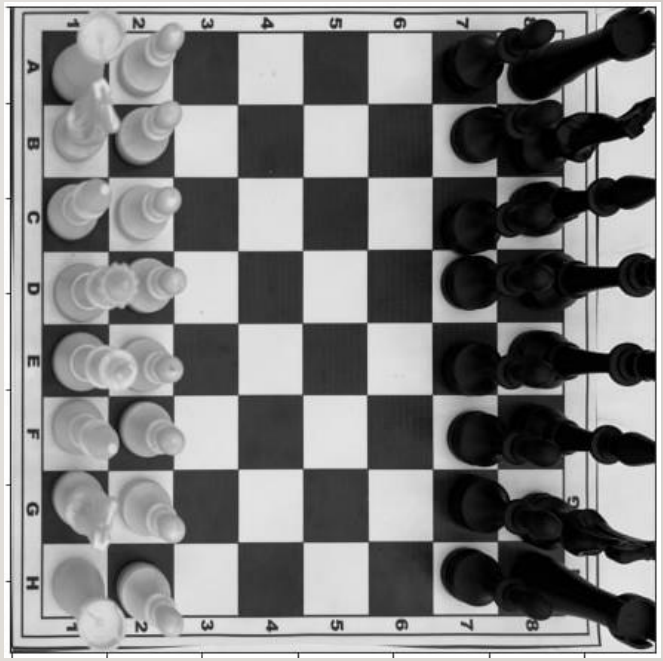
Doesn't work well at extreme angles due to poor projective transformation.The figures are very elongated and overlap each other. It is possible to train the classifier on these data, but due to the small amount, this could not be done.

Quality estimation

We used the F1-score metric to evaluate the quality of our model. It is extremely important that all pieces are found and classified correctly, otherwise we will get a wrong position and chess algorithms will not work correctly.

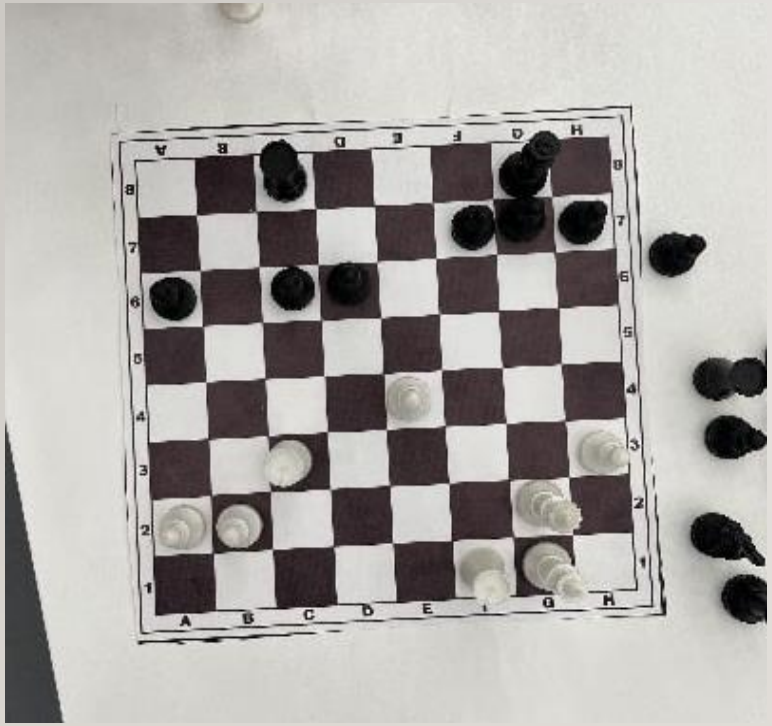


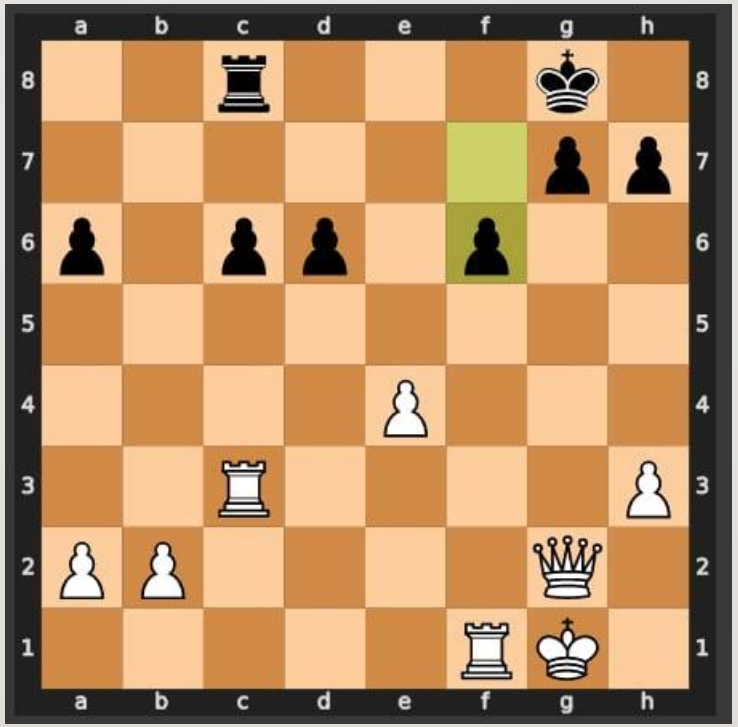
F1-score = 1



F1-score = 0.66

Example of prediction

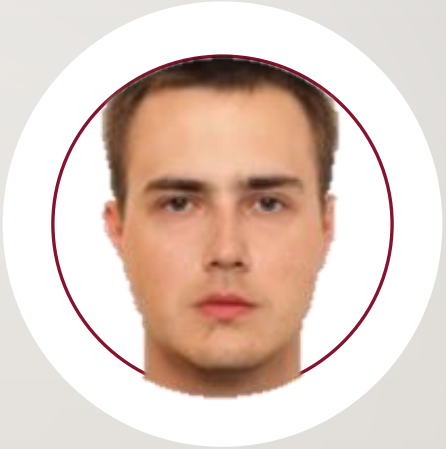




We can see correct position and we plotted the top-1 move on virtual board.

Our team

Nikolay Ivanov



Bogdan Alexandrov

Thank you for attention!