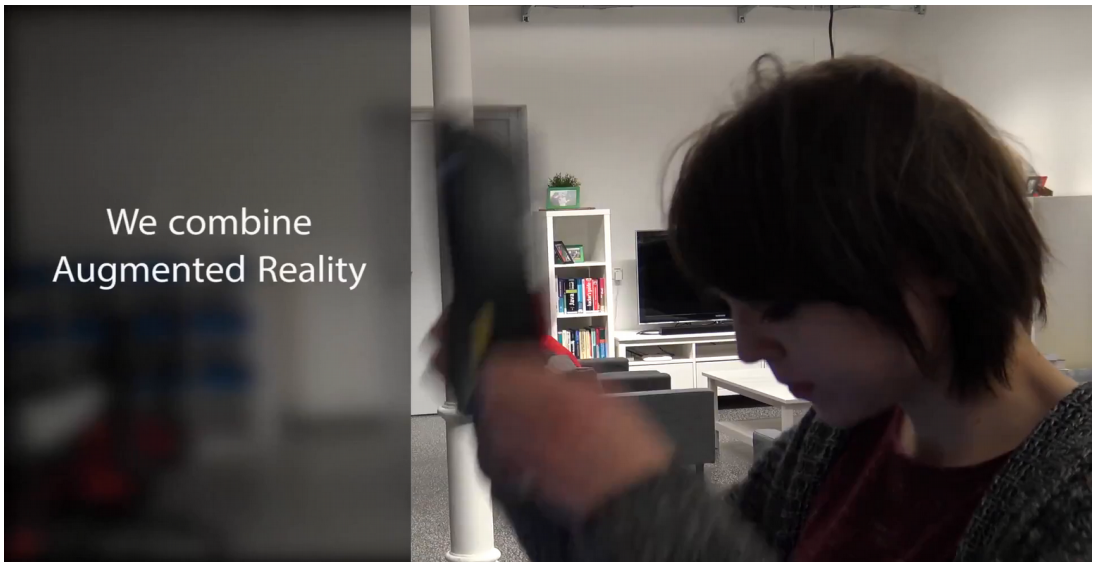
Chess 2.0

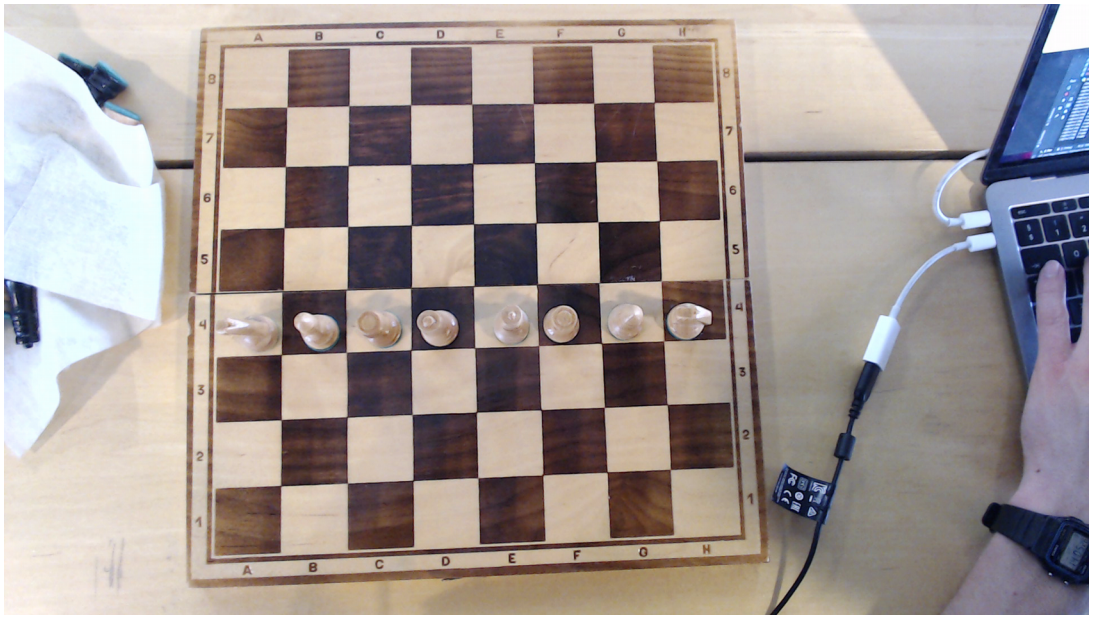
Casper V. Kristensen
Alexander Munch-Hansen

Idea



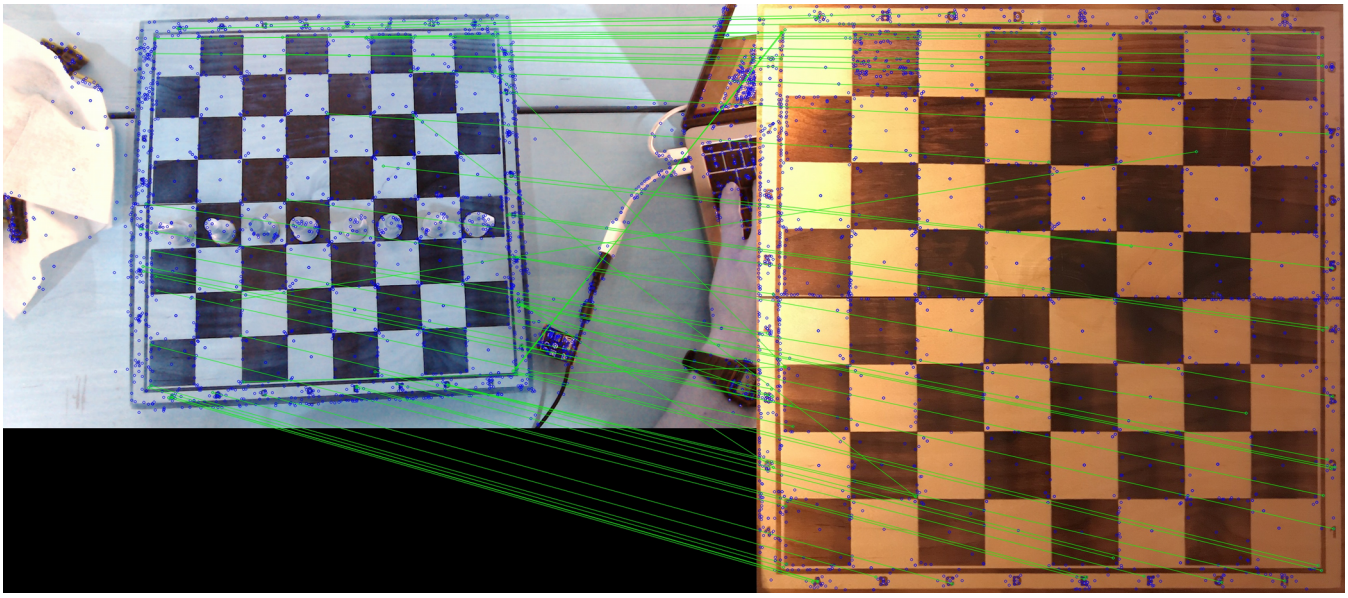
https://www.youtube.com/embed/Geyr95Nl8mc?start=4&end=23&autoplay=1

Camera Image



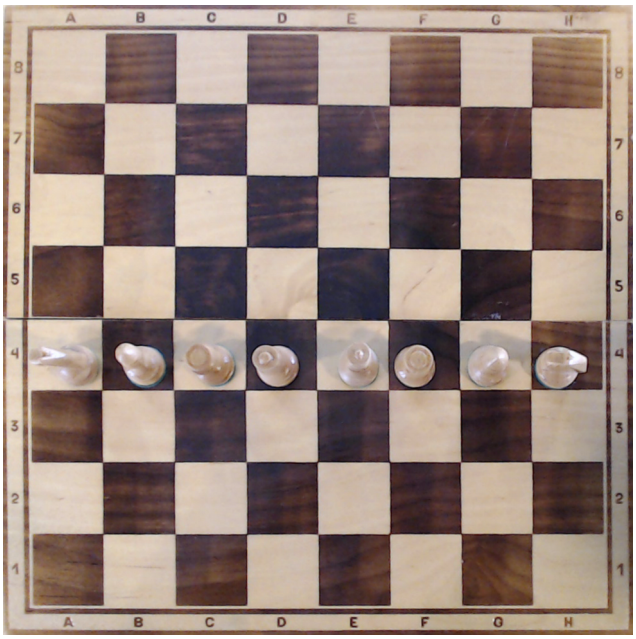
Baseline Board

Matches

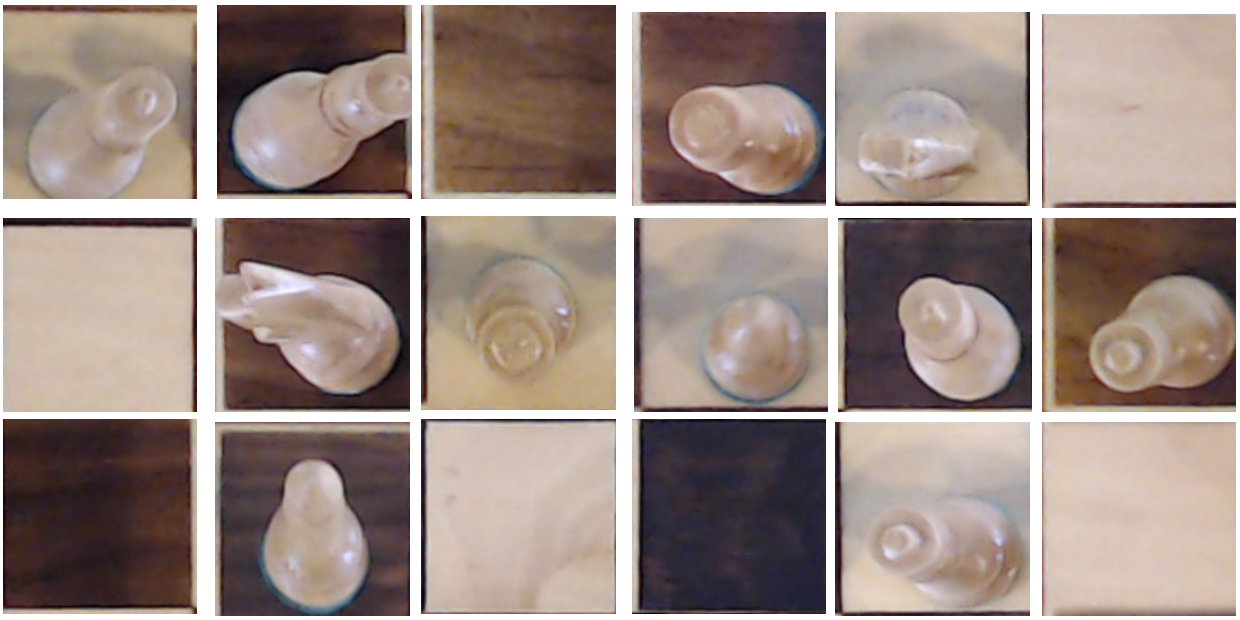


1) Baseline board.
2) FLAN feature matching in B/W.
3) RANSAC outlier detection.
4) Compute homography on these matches.

Warped Board

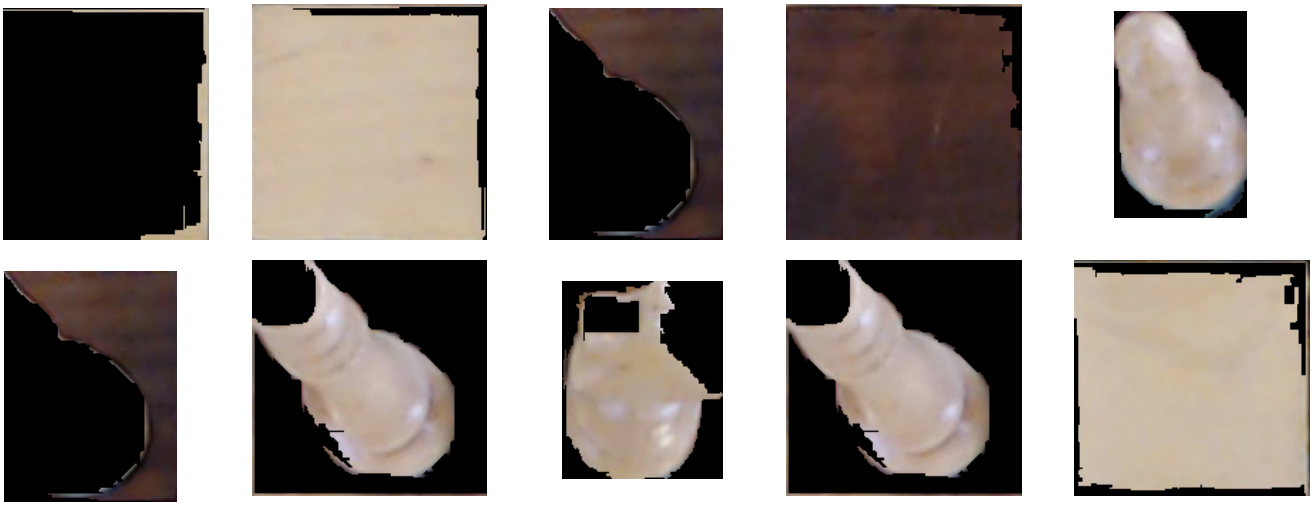


Squares



- Split board in 8x8 squares.
- Note that the same piece can look very different on A1 compared to H8.

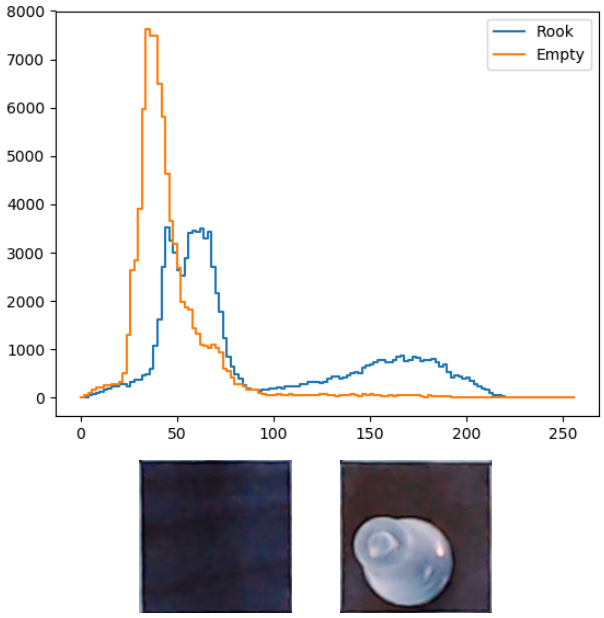
Segment Detection

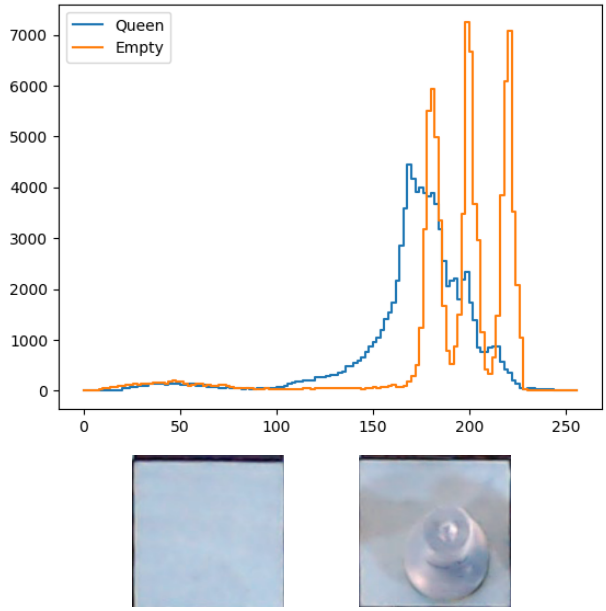


Largely based on heuristics.

- Crop square around center to remove inaccurate warp, i.e. edge of next square visible.
- Compute segments.
- Occupied square implies at least two segments: 1) the square 2) the piece.
- Remove extreme values:
  - Too large a segment is the entire square
  - Too small a segment is the edge of the adjacent square.

Histograms



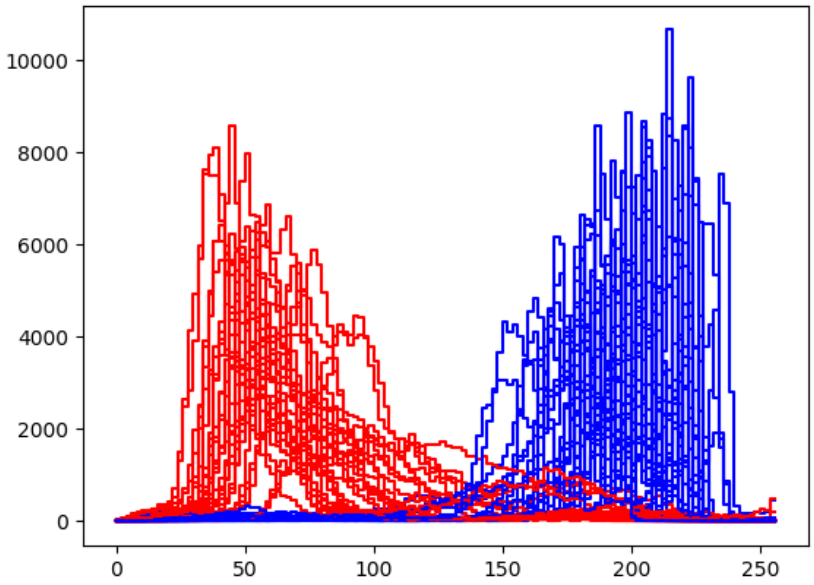


- Segment algorithm removes most of the empty squares, yet false negatives remain.
- It seems possible to distinguish empty and non-empty squares in the histograms.
- Feed this data to SVM.
- Since we already know the square's position, we know the colour of the square:
  - Train two different SVMs.

Sidenote: Our wooden board introduces large variance in the squares, e.g. veins of differing colour.

Future work: Overfit to our particular board by training 64 individual SVMs – one for each square.

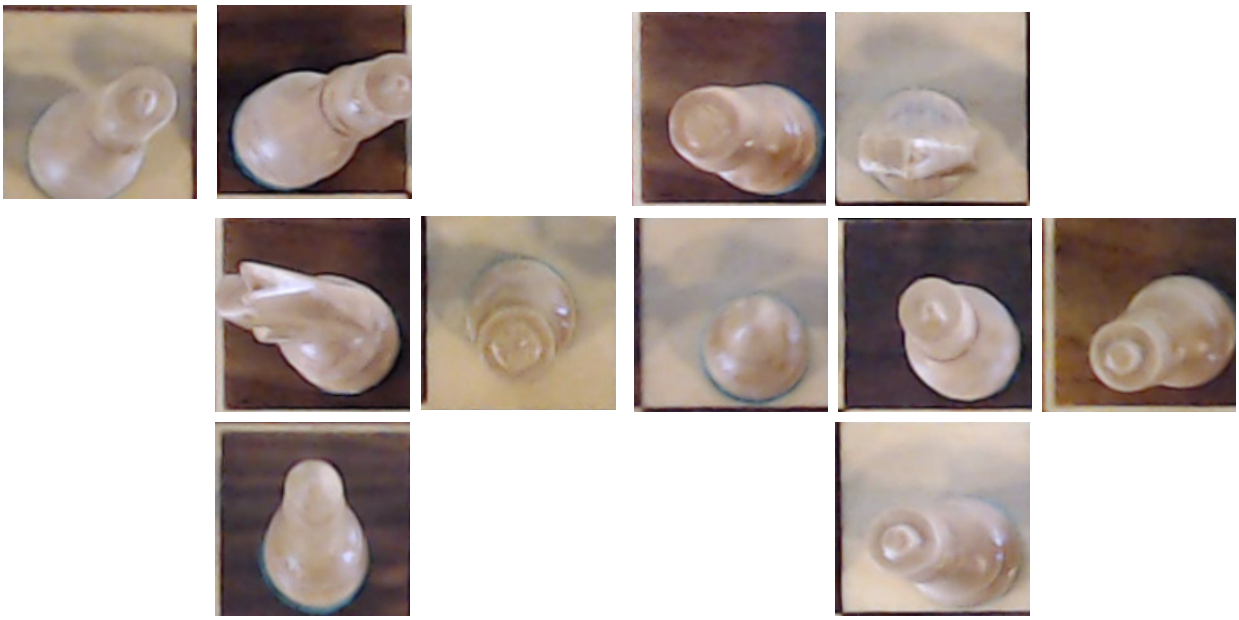
Histograms Combined



Why we use a different SVM based on the square colour.

Red: Black square
Blue: White square.

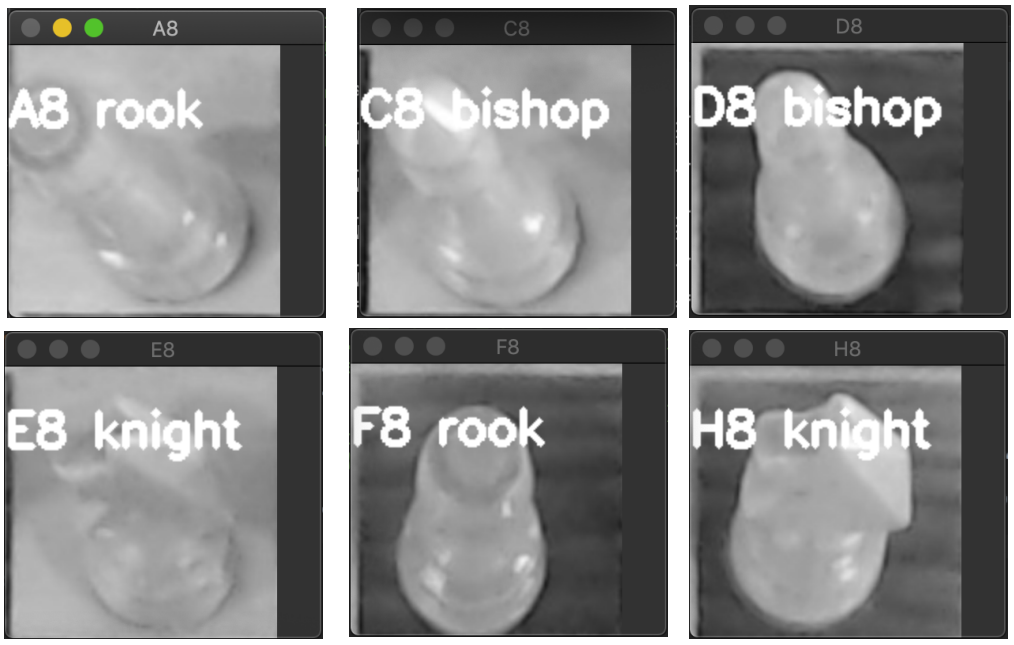
Occupied Squares



- Successfully removed all empty squares.

- We can now utilise Neural Network to classify the pieces.

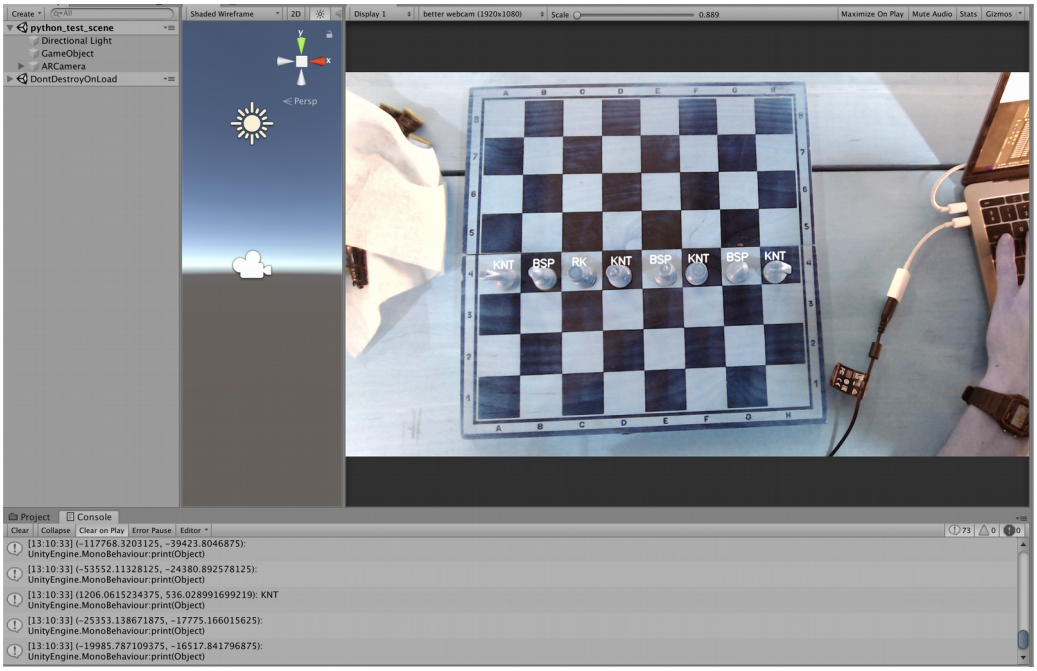
Neural Network Classification



Why not use a neural network from the beginning?

- We found that NNs were good at finding features in the empty squares (because of the glossy texture of our board, very vissible veins in the tree etc.), and thus misclassifying empty squares as pieces.
- By first removing empty squares, these problems are avoided.

Unity


Create
python_test_scene
Directional Light
GameObject
ARCamera
DontDestroyOnLoad
Shaded Wireframe
2D
Persp
Display 1
better webcam (1920x1080)
Scale
0.889
Maximize On Play
Mute Audio
Stats
Gizmos
KNT
BSP
RK
KNT
BSP
KNT
BSP
KNT
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Editor
[13:10:33] (-117768.3203125, -39423.8046875):
UnityEngine.MonoBehaviour:print(Object)
[13:10:33] (-53552.11328125, -24380.892578125):
UnityEngine.MonoBehaviour:print(Object)
[13:10:33] (1206.061523437 5, 536.028991699219): KNT
UnityEngine.MonoBehaviour:print(Object)
[13:10:33] (-25353.138671875, -17775.166015625):
UnityEngine.MonoBehaviour:print(Object)
[13:10:33] (-19985.787109375, -16517.841796875):
UnityEngine.MonoBehaviour:print(Object)

Architecture