2D Chess Table

Student: **Boga Titus**, gr. 30433

Supervisor: Robert Varga

25 may 2022

# Introduction

The “2D Chess Table” project represents a way of recognizing and interpreting of an image containing on certain positions smaller images given as templates. For this project the bigger image represents a Chess Table and the smaller ones represent the pieces and the free spaces.

The motivation behind the project is to be able to transform the given information in usable computer usable data which. If by processing an image, we can understand the information given in the certain Chess picture then usages of such a software can become very useful and impactful in the given domain.

# State of the art

While considering the usage of OpenCV there could be multiple approaches taken for solving such a problem. One of which could be by detecting the edge of an “object” a piece and the check if it resembles one of the template pieces which wouldn’t be that straight forward to implement. Another approach could be to look for the pieces in the expected places and again compare them to the template pieces.

# Proposed method

My approach for this problem is to take an image (board) containing the 8x8 places which could contain either a piece which should also be given as template or an empty space. In order to process the image and transform it in a standard format we would check every place and compare it with a template piece.

The comparation between a place from the map and a piece is done by iterating pixel by pixel through the board and a piece and check for equality. If they are indeed identical then we would mark in a matrix this information.

Diagram

Description automatically generated

*Fig 1. Block diagram*

# Experimental results

The input for this algorithm should be a png image the size of 64 x a piece size and it can be tested by using proper images (which can be generated from the code). Testing the software can be done by using different placements for the given pieces and expect the algorithm to show a standardized output for the given image. Below is an example of an image (.png).

A picture containing text, silhouette

Description automatically generated

*Fig 2. Example of input image*

For this image the output of the algorithm will look like this:

A picture containing text

Description automatically generated

*Fig 3. Output of the image from Fig 2*

# Conclusions

After working in this project there came quite a few achievements such as more experience working with OpenCV and understanding some more of the technologies used for this project. Another big achievement is the fact that the algorithm correctly detects and interprets the found templates from a given image managing to work very well for a correct input.

Considering the pros and the cons of the chosen approach I have to mention that one of the big advantages would be the fact that the implementation is straight forward and easy to understand and adapt. However, the limitations are seen when introducing imperfect input, where the algorithm may struggle with the detection.

As future developments I would like to fix the known issues by making the algorithm more flexible around different type of input images so that it would detect the template pieces from a more complex input.

# Bibliography

* http://users.utcluj.ro/~robert/ip/ip.html