Documentation

Double Double Dominoes score calculator

Regular task

In order to work with very similar images we firstly have to extract the board from the images containing different translations, rotations, scaling and perspectives. For this we will need a template of the board which we create by cropping the straightest image of the empty board (the one where all sides are parallel to the *Ox* and *Oy* axes). We chose the image *board+dominoes/02.jpg* and cut a square image containing only the board (Figure 1).

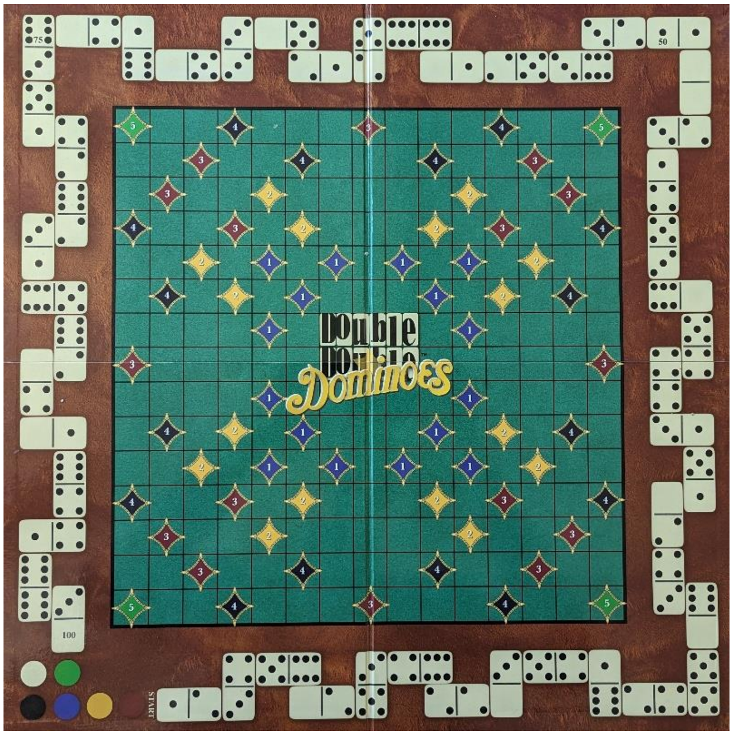

Figure 1 – template

We now use SIFT descriptors, Lowe’s ratio test and the RANSAC algorithm to create the homography matrix that will transform every image (Figure 2) so that we get one where the board aligns perfectly with our template (Figure 3).

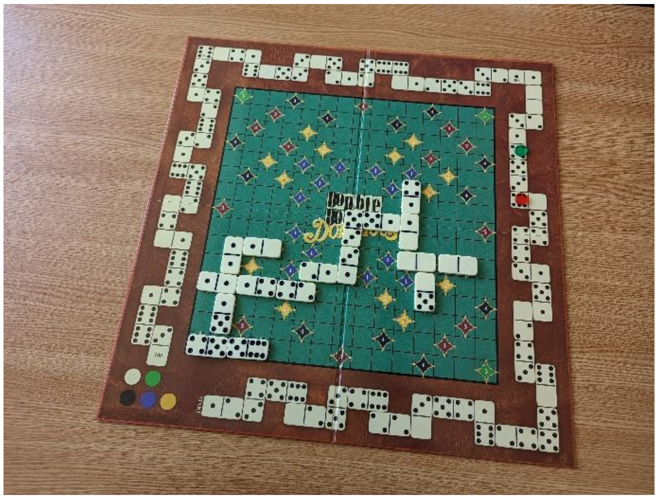

Figure 2 – perspective image

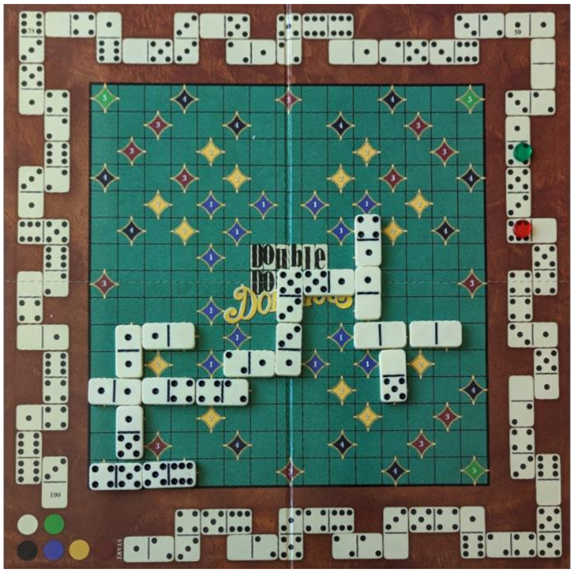

Figure 3 – extracted board

Because we now know that the coordinates of every detail of the images will be the same ones as in the template we can identify the *x* coordinates of the 16 vertical lines and the *y* coordinates of the 16 horizontal lines of the playing area.

In order to identify on which positions new dominoes were added we use the mean squared differences to compare the current image with the previous one (or with the template if it is the first move of the game). Due to the rules of dominoes, at each move we firstly look at the patches that already have a domino next to them and choose the one with the biggest difference to itself in the previous image (or directly 8H if it is the first move). Now that we have one half of the new piece we identify the other half by looking only at the empty neighbors of the first half and choose the one with the biggest difference. For each patch we subtract 5 pixels from each side to prevent cases where dominoes were placed slightly crooked.

For the classification of the numbers on the dominoes we used template matching. We created 10 binary templates (one for each of 0, 1, 4, 5 and two for 2, 3, 6 as they are different depending on the orientation) (Figure 4).

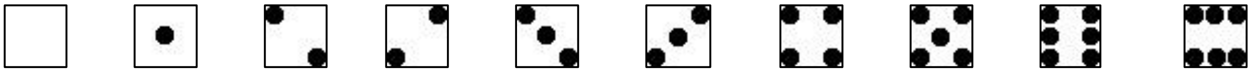

Figure 4 – domino templates

After identifying the positions on which a new domino was added we select each half adding 5 pixels to each side to make sure we have it whole, transform it to grayscale and then to binary with a threshold of 150 (Figure 5). We then perform template matching on it and attribute it the number of the template with the highest similarity score.

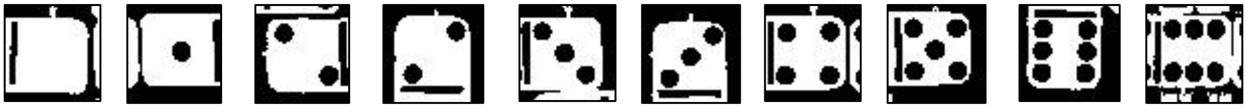

Figure 5 – patches to classify

Finally, for the score we hard-code two lists, one with the score of the diamonds and one with the score track. At each move we calculate the score according to the rules by also keeping track of the total score for each player.

Bonus task

For the bonus task the board extraction and domino classification is done exactly the same. The only difference in order to obtain the configuration is that we now have all the pieces placed on the board. Therefore, we will choose a threshold (4750) for the mean squared differences above which we consider a position occupied.

After computing a matrix of the configuration of the board we divide it into domino pieces by applying some rules that assure us that a position is part of a new piece. We run this algorithm until no positions that satisfy any of these conditions exist.

Finally, we iterate through the two matrices (the one with the configuration and the one with the piece separation) and if we find a 2x2 square that contains dominoes or two adjacent pieces whose ends don’t match then we have a mistake.