University of Bucharest
Faculty of Mathematics and Computer Science
Department of Computer Science

Alexandra Diaconu
Ciprian-Mihai Ceauşescu
Bogdan Alexe
MSC in Computer Science
April 2023

Computer Vision - Project 1

Double Double Dominoes score calculator

Objective

The goal of this project is to develop an automatic system for scoring a specific variant of dominoes game called Double Double Dominoes.

Dominoes

Dominoes[1] is a family of tile-based games played with gaming pieces. Each domino is a rectangular tile, usually with a line dividing its face into two square ends. Each end is marked with a number of dots (usually for 1 to 6) or is blank.

Double Double Dominoes

Double Double Dominoes (DDD) is a specific variant of dominoes game that combines ideas from the game of a Scrabble into a dominoes game (Figure 1). The object of the game is to score points by building chains of dominoes that cover the diamonds on the board. The farther you get from the center, the bigger the score is. The score can be increased by

[1] https://en.wikipedia.org/wiki/Dominoes

Figure 1: *The DDD board (left) and all possible domino tiles placed on the board (right).*

using double dominoes (domino tiles with the same value at both ends, such as 0-0, 1-1, etc) or taking advantage of the position of the score piece placed on the track score.

Board

The DDD playing board (Figure 1) is divided into a 15×15 grid of squares and the score track that occupies the outside border of the game board. The board is marked with squares containing numbered colored *diamonds*:

- 16 blue diamonds for 1 point;
- 16 yellow diamonds for 2 points;
- 16 brown diamonds for 3 points;
- 16 black diamonds for 4 points;
- 4 green diamonds for 5 points.

The play

The first player must place the domino on the star $\star$ square at the center of the board. Every other domino must be placed so that one end touches exactly one domino with the matching number. The other end either touches nothing or bridges the gap by matching another domino. When a player plays a double domino, the player may choose to immediately place another domino onto the board but it must be placed next to that double domino. The player turn ends when he chooses not to place any additional dominoes, or the player places a domino that is not double domino. Figure 2 shows valid and invalid configurations on a general domino game board.

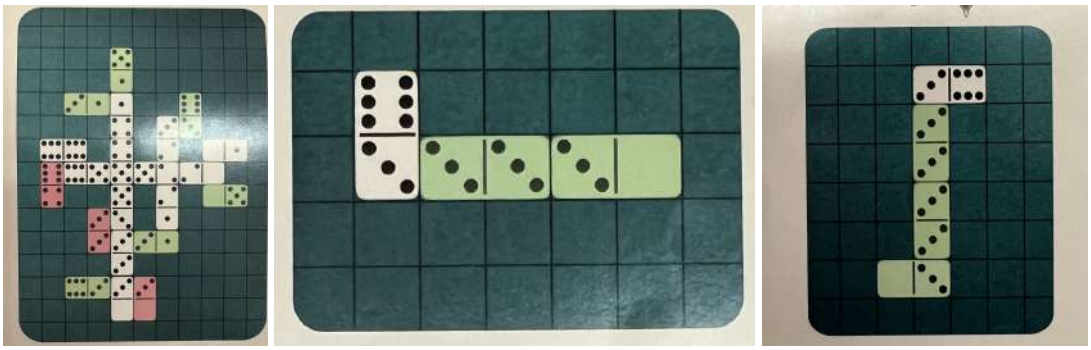

Figure 2: *The white dominoes in the diagrams represent dominoes on the board at the beginning of a turn. The green dominoes in the diagrams show various ways to correctly place new dominoes.* **Left***: the red dominoes show incorrect placement of new dominoes.* **Middle***: placing a double domino and adding a second domino that matches the head of the double domino.* **Right***: placing a double domino, then adding a second double domino, then adding a third domino that matches the head of the second double domino.*

Scoring

The score track occupies the outside border of the game board. Whenever a player scores points, immediately it should move his score piece on the score track a number of spaces equal to the number of points scored. There are two ways to score points:

- placing a domino on the colored diamonds on the board. When a player places a domino that covers a square containing a numbered diamond, the player scores the number of points indicated by the number in the diamond. If the player plays a double domino, then the player scores double the points.
- every time a domino is placed on the board such that it matches the number occupied by a score piece on the score track by a player, that player scores 3 bonus points. For example, if your score piece is on a space with 4 dots, and you or another player place a domino on the board whose tail or head has 4 dots, you move your score piece 3 spaces ahead. Bonus points are only scored once for each domino played.

For better understanding of the playing and scoring rules you can have a look at this video: https://www.youtube.com/watch?v=ORPJIjFkluQ&t.

Scoring example

While, in general, a player can make many moves in a turn (by placing a double domino and then another additional domino and then possibly another domino - see Figure 2) we consider in our project only individual moves (and not turns). At each move, we specify which player makes the move. We also consider the scenario with only two players: Player1 has a red score piece and Player2 has a green score piece. Figure 3 shows seven moves made at the beginning of a game. We list below the seven moves and offer detailed explanations about computing the corresponding score of each player after each move. At the beginning of the game, both score pieces (red and green) are in the left bottom corner, at start, they do not cover any space with any dots (0 to 6).

Move 1. Player2 starts the game by placing the 2-1 domino tile. Neither player scores any points after the first move. Both score pieces are at start. The next move is executed by Player1.

Move 2. Player1 places the 2-2 double domino tile. Neither player scores any points after this move. Both score pieces are at start. Player1 has the chance to continue with another move (as he previously placed a double domino) but he stops as he doesn't have a tile to continue so the move 3 is executed by Player2.

Move 3. Player2 places the 2-1 domino tile covering a blue diamond with value 1 so Player2 scores 1 point. Consequently, Player2 moves his green score piece on the score track on the first space which has value 1 (one dot). The red score piece of Player1 is at start. The next move is executed by Player1.

Move 4. Player1 places a 2-3 domino tile covering a blue diamond with value 1 so Player1 scores 1 point. Consequently, Player1 moves his red score piece on the score track on the

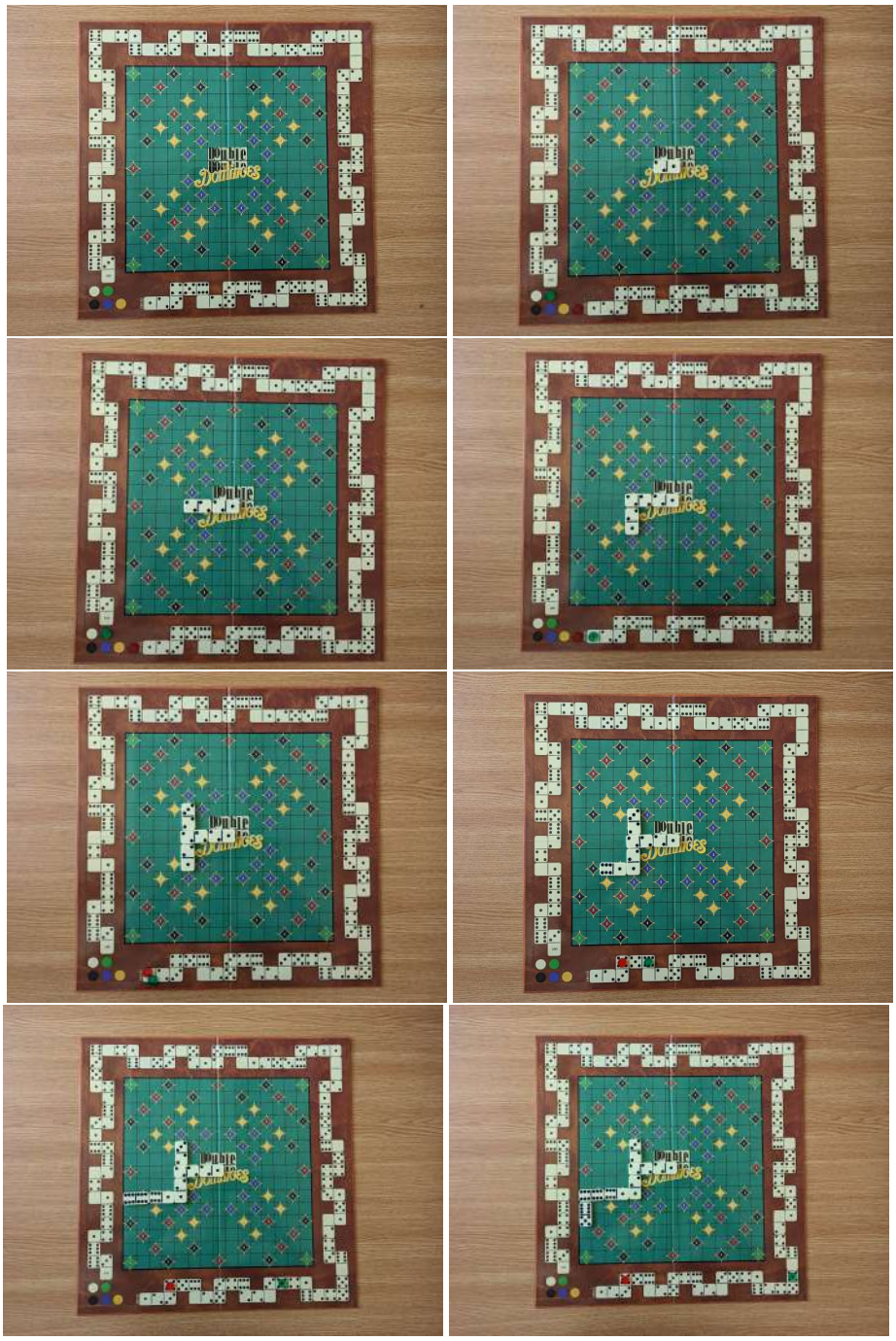

Figure 3: *We illustrate seven moves made at the beginning of a game, starting from the empty board. We describe in detail in the text the computation of the corresponding scores to each of the two players after each move.*

first space which has value 1. Notice that both pieces, red and green, are positioned at the same place on the score track. The next move is executed by Player2.

Move 5. Player2 places a 1-6 domino tile covering a yellow diamond with value 2, so Player2 scores 2 points. As Player1 and Player2 have their score pieces placed on the score track at a place with value 1 and a domino containing an end with value 1 was placed in the current move, this means that each of the players get another additional 3 bonus points. So, Player2 advances 5 spaces on the score track (2 points for placing a domino on the yellow diamond and 3 points for placing a domino with 1) and Player1 advances 3 spaces. Now, the green score piece of Player2 covers a space with value 6 and the red score piece of Player1 covers a space with value 4. The next move should be executed by Player1 but Player1 passes as he doesn't have any good option to play a domino tile. So next move is executed again by Player2.

Move 6. Player2 places a 6-6 double domino tile covering a black diamond with value 4, so Player2 score 8 points = 4 points × 2 (as he placed a double domino). Player2 also has his score piece positioned on the score track at a place with value 6 so he scores additional 3 bonus points. After this move, the red piece score of Player1 remains unmoved at the place covering value 4, while the green piece score of Player2 has moved 11 positions ahead, reaching a place with value 5. As Player2 placed a double domino he has the chance to continue moving. The next move is executed by Player2.

Move 7. Player2 places a 6-5 domino tile covering a black diamond with value 4, so Player2 score 4 points. Player2 also has his score piece positioned on the score track at a place with value 5 so he scores additional 3 bonus points. After this move, the red piece score of Player1 remains unmoved at the place covering value 4, while the green piece score of Player2 has moved 7 positions ahead, reaching a place with value 5.

Data description

The release data directory (available here https://tinyurl.com/CV-2023-Project1) contains four directories: *board+dominoes*, *train*, *test* and *evaluation*. All directories contain images taken with Bogdan's phone in different scenarios. The directory *board+dominoes* contains several images with the empty board from different viewpoints and also with the board on which we placed all possible 28 dominoes tiles (from 0-0, 0-1 to 6-6). You can use these images to better understand the problem and to extract data for your solution. The directories *train* and *test* have the same structure, although the *test* data will be made available after the deadline. The *train* directory contains two directories: *regular_tasks* and *bonus_task*. The regular tasks folder contains data for 5 games, where each game has 20 moves. In total there are 100 images and 100 annotations files. For each game, there is a .txt file that annotates which player makes a move in the respective round (*g_moves.txt* where g is the game number). The training image i corresponding to game g is denoted by the file 'g_i.jpg', where $g \in \{1, 2, 3, 4, 5\}$ and $i \in \{01, 02, 03, ..., 20\}$. The corresponding annotation file has the extension '.jpg' replaced by '.txt'. We describe the bonus task later.

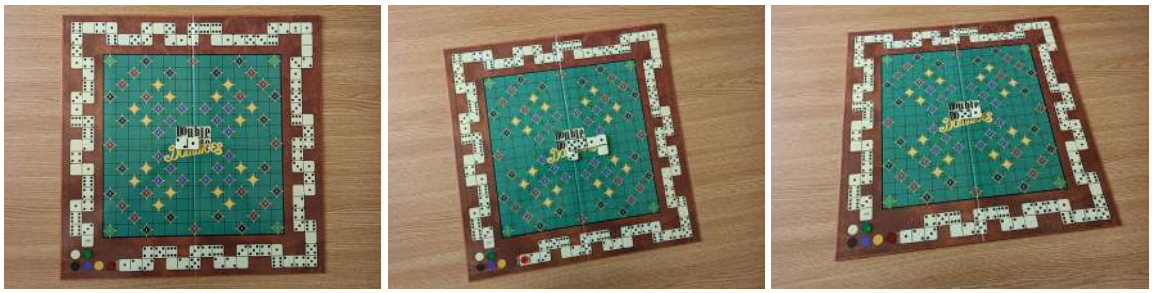

Figure 4: *Different views of the board: regular view (left), rotated view (centre) and perspective view (right).*

The first two games consists of images with regular views, the images are taken with the phone placed parallel above the board, with minor scale changes and rotations. The third game consists of images with rotated views, the images are taken with the phone placed parallel above the board, but at some non negligible rotation with respect to the board. The fourth game consists of images with perspective views, the images are taken with the phone tilted with respect to the board. The fifth game consists of images with mixed views, either regular, rotated or perspective. Figure 4 illustrates these scenarios.

The ground-truth annotations files contain for each move the following information:

- the position of the domino tile added to the board, taken in the order from left to right and from top to down. We specify the position using numbers 1-15 for rows and letters A-O for columns. Take into consideration that only one domino is placed at each move.
- the number of dots (from 0 to 6) on the domino tile at the corresponding position on the board.
- the total score obtained in the current move by the player that makes the move.

Figure 5 exemplifies a ground-truth annotation file for move 5 presented earlier.

The directory *evaluation* shows how the evaluation will take place on the test data after the deadline. It contains the following subdirectories:

- *fake_test* - this directory exemplifies how the test data will be released, keeping the structure of the previously described *train* directory. Notice that we include here only 1 game = 20 images for regular tasks and 5 images for bonus task. The test data will contain 5 games = 100 images for the regular tasks and 20 images for bonus task.
- *submission_files* - this directory exemplifies the format of the results data that we expect from you to submit in the second stage. You will have to send your results in this format, uploading a zip archive of a folder similar with the one called *Alexe_Bogdan_407*. Notice that the current archive correspond to the data released in the *fake_test* directory. For the test stage your archive should contain 100 files for the regular tasks and 20 files for bonus task.

Figure 5: *For the current move, a domino with 1-6 is placed on the board. The ground-truth annotation file specifies the positions of newly added domino, the numbers of the domino and the total score of the player in the current turn.*

- *code* - this directory contains code that we will use to evaluate your results using the ground-truth data. Make sure that this code will run on your submitted .txt files. The ground-truth data will be released after you send us your results.

Very important. There is a small difference between train and test data: only in the training images you have the score pieces placed on the score track, in the test images these score pieces will not be present (actually it is your task to find out the position of these score pieces on the score track). Figure 6 exemplifies the difference between train and test images.

Figure 6: *Train image with score pieces (left), test image without score pieces (right).*

Bonus Task

You can get additional 0.5 points by correctly solving the bonus task. This task consists in localizing the incorrect placements of new dominoes (Figure 2). Incorrect placements of new dominoes can appear when:

- a domino touches more than one domino. In this case a square 2×2 will be formed on the board;
- a domino touches the same domino at both ends. In this case a square 2×2 will be formed on the board;
- a domino touches two different dominoes but does not bridge a gap. In this case a square 2×2 will be formed on the board
- a domino doesn't match the head or end of the domino that is attached to.

In this task you will get 10 images where there are some incorrect placements. For this task you need to count the number of incorrect placements and localize the position of them. In Figure 7 you can see the incorrect placements and how we annotate them. For every image in which you solve this task correctly you get 0.05 points.

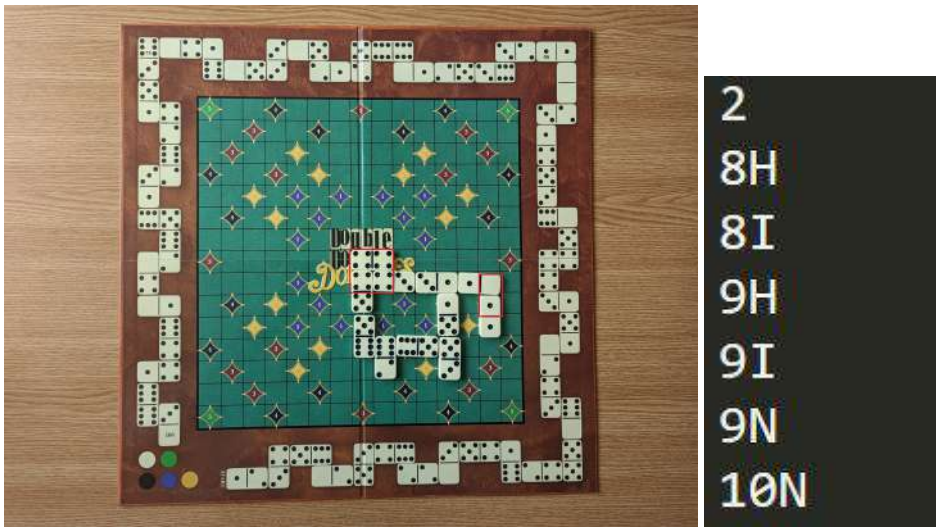

Figure 7: *There are two incorrect placements: (1) placing the domino 4-4 on positions 8H and 8I that connects with both ends at 9H and 9I forming a* 2×2 *square; (2) placing the domino 1-1 on positions 10N and 11N that doesn't match the 1-0 domino at position 9M and 9N. We annotate all the positions that create conflict.*

Requirements

Your job is to write a program in Python/Jupyter notebook that automatically solves the task of extracting information of the current move depicted in a test image. For each test image you have to output the corresponding information similar to the annotation files,

thus specifying the position of the newly added domino to the board, the number of dots (from 0 to 6) that appear on the newly placed domino at the corresponding position on the board and the total score obtained in the current move by the current player.

This project worths 5 points but you can gain a bonus by solving the bonus task (0.5 points) for a total of 5.5 points. We will grade your project based on the performance achieved by your algorithms on each of 110 test images (100 images for regular tasks and 10 images for bonus task).

For regular tasks, you will receive a test set containing 100 testing images organized in 5 games of 20 moves. The distribution of images in the test data follows the distribution of train data, meaning that the images were acquired in the same conditions and also that the first two games will contain images with regular views, the third game will contain images with rotated views, the fourth game will contain images with perspective view, the fifth game will contain images with mixed views. For each test image you have to output a .txt file containing information similar to the annotation files. Each correctly solved test image worths 0.045 points. For correctly specifying the two positions of the added domino tile you receive 0.015 points per image (move), for correctly specifying the two values (number of dots) on the domino tile placed at the corresponding positions on the board you receive 0.015 points per image (move) and for correctly specifying the total score of the current player you receive 0.015 point per image (move). You receive 0.5 points from *ex officio* conditioned on the fact that you respect the format of the submitted results, such that our evaluation script works smoothly on your provided results.

Deadlines

Submit a *zip archive* containing your code (python files or Jupyter notebook files), all auxiliary data that you are using (templates, models, etc.) and a pdf file describing your approach until Tuesday, 2^{nd} of May using the following link https://tinyurl.com/CV-2023-PROJECT1-SUBMISSIONS. Please do not include in your zip archive any unuseful data (like training images, we already have them!!!). Notice that this is a hard deadline, no projects will be accepted after the deadline. Your code should include a README file (see the example in the materials for this project) containing the following information: (i) the libraries required to run the project including the full version of each library; (ii) indications of how to run the solution and where to look for the output file. Students who do not describe their approach (using a pdf file) will incur a penalty of 0.5 points.

On Wednesday 3^{rd} of May we will make available the test data. You will have to run your solution on the test images provided by us and upload your results in the same day as a zip archive using the following link https://tinyurl.com/CV-2023-PROJECT1-RESULTS.