SE 3XA3: Module Interface Specification
Sudoku Solver

Team 08, SudoCrew
Rashad A. Bhuiyan (bhuiyr2)
Kai Zhu (zhuk2)
Stanley Chan (chans67)

March 18, 2022

Contents

1 Module Hierarchy

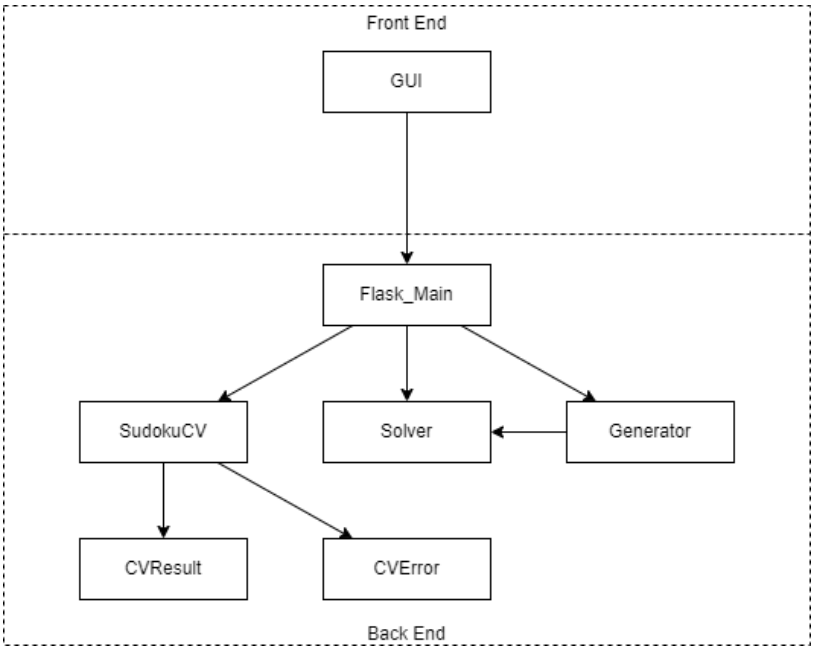

Figure 1: Use hierarchy among modules

2 SudokuCV Module

2.1 Syntax

2.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
__init__	string		NO_MODEL
recognize	image, boolean, boolean	CVResult	NO_GRID, GRID_SMALL, IMG_SMALL

2.2 Semantics

2.2.1 State Variables

model: The Keras machine learning model loaded for this instance of the SudokuCV.

2.2.2 Environmental Variables

model_file: file path to a trained Keras machine learning model used for digit recognition.
dimensions = (900, 900): specifies resolution of intermediate images during processing
min_dimensions = (200, 200): specifies minimum resolution required to process the input image

2.2.3 Assumptions

__init__ is called before any other access routine

2.2.4 Semantics of Exported Routines

- **__init__(model_file)**
  Input:
  * model_file: string; path to the trained machine learning
  Transition:
  * model_file is loaded into model state variable using Keras
  Exceptions:
  * NO_MODEL, if no valid model file is found at model_path
- **recognize(image, is_file, show_image)**
  Input:
  * image: a source image file path or buffer string, depending on is_file
  * is_file: optional boolean (default=True), indicates whether source is file or buffer data
  * show_image: optional boolean (default=False), displays intermediary processed images for debugging
  Output:
  * CVResult object containing recognition results processed from input image.
  Exceptions:
  * NO_GRID, if no square grid is detected in the input image.
  * GRID_SMALL, if no square grid is detected in the input image.
  * NO_GRID, if no square grid is detected in the input image.

3 CVResult Module

3.1 Syntax

3.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
__init__	int[], float[], image, string		
getConfidence		2D float[]	CV_FAIL
getConfidentResults	float	2D int[]	CV_FAIL
saveImg	string		CV_FAIL, SAVE_FAIL
deleteImg	string		CV_FAIL, DEL_FAIL
getImage		image	CV_FAIL

3.2 Semantics

3.2.1 State Variables

uuid: unique identifier for this CVResult object
raw_results: integer array of recognized digits
confidence: float array of prediction confidence for each recognized digit
error: error message string, empty if recognition did not experience error
image: a processed image of the original input

3.2.2 Environmental Variables

N/A

3.2.3 Assumptions

__init__ is called before any other access routine.
Error message string is passed into the constructor if recognition failed.

3.2.4 Semantics of Exported Routines

- **__init__(raw_results, confidence, image, error)**
  Input:
  * raw_results: integer list of recognized digits with 0 representing empty
  * confidence: float list of confidence rate from 0 (no confidence) to 1 (certain) for each result
  * image: processed image of the original input
  * error: error string description of exception encountered during recognition

  Transition:
  * raw_results and confidence lists are converted to numpy arrays and saved in respective state variables.
  * image and error are stored in state variables
  * a UUID (universally unique identifier) is generated and stored in the uuid variable

  Exceptions: N/A
- **getConfidence()**
  Input: N/A
  Output:
  * 2D float list representing confidence rates (0 to 1) for each recognized cell content

  Exceptions:
  * CV_FAIL, if error state variable is not empty
- **getConfidentResults(threshold)**
  Input:
  * threshold: float value to cut off prediction acceptance at

  Output:
  * 2D int list of recognized digits with confidence above threshold, 0 otherwise

  Exceptions:
  * CV_FAIL, if error state variable is not empty
- **saveImg(directory)**
  Input:
  * directory: string path of the directory to save the image in

  Transition:
  * The image is saved in the specified directory using the UUID as file name

  Exceptions:
  * CV_FAIL, if error state variable is not empty
  * SAVE_FAIL, if directory does not exist or permission is denied
- **deleteImg()**
  Input:
  * directory: string path of the directory to delete the image from

  Transition:
  * The image with the UUID as file name is deleted from the directory

Exceptions:
  * CV_FAIL, if error state variable is not empty
  * DEL_FAIL, if file does not exist or permission is denied

- **getImage()**
  Input: N/A
  Output:
  * processed image encoded as a base64 string
  Exceptions:
  * CV_FAIL, if error state variable is not empty

4 CVErrors Module

4.1 Syntax

4.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
getErrorMessage getErrors	int	string dictionary	UNDEFINED

4.2 Semantics

4.2.1 State Variables

N/A

4.2.2 Environmental Variables

ERR_MSGS: dictionary of error code matched with string descriptions

4.2.3 Assumptions

N/A

4.2.4 Semantics of Exported Routines

- **getErrorMessage(errorID)**
  Input:
  * errorID: int error code
  Output:
  * string description of the error
  Exceptions:
  * UNDEFINED, if errorID is not in the ERR_MSGS dictionary
- **getErrorList()**
  Input: N/A
  Output:
  * returns all available error key-value pairs in the ERR_MSGS dictionary
  Exceptions: N/A

5 Solver Module

5.1 Syntax

5.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
solve	2D int[]	boolean	
valid	2D int[], int[], int	boolean	INVALID_POSITION, INVALID_NUM
find_empty	2D int[]	int[]	
getSolvedCoordinates	2D int[]	int[]	

5.2 Semantics

5.2.1 State Variables

N/A

5.2.2 Environmental Variables

N/A

5.2.3 Assumptions

Input Sudoku board is a valid 9x9 2D array containing integers valued 0 to 9.

5.2.4 Semantics of Exported Routines

- **solve(board)**
  Input:
  - board: 2D 9x9 integer list containing digits 0 to 9, where 0 represents empty space

  Transition-Output:
  - If solvable, the board is solved in place, where all 0s in the array are replaced by solution digits. The method returns True.
  - Returns false if the board is not solvable.

  Exceptions: N/A
- **valid(board, position, number)**
  Input:
  - board: 2D 9x9 integer list representing the cells on the game board
  - position: integer tuple (row, column) to specify a position on the board
  - number: the integer digit to test for validity at the specified position

  Output:
  - True, if number at specified position does not violate Sudoku game rules
  - False, if otherwise

  Exceptions:
  - INVALID_POSITION, if for i = 0..1 $position[i] > 8$ or $position[i] < 0$
  - INVALID_NUM, if $number > 9$ or $number < 1$
- **find_empty(board)**
  Input:
  - board: 2D 9x9 integer list representing the cells on the game board

Output:
  * (row, column) integer tuple of the position of the first empty cell on the input board.
  * returns empty tuple if no free cells found

Exceptions: N/A

- **getSolvedCoordinates(board)**

  Input:
  * board: 2D 9x9 integer list representing the cells on the game board

  Output:
  * list of integer tuples with (row, column) indicating positions of empty cells to be solved

  Exceptions: N/A

6 Generator Module

6.1 Syntax

6.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
generateRandomValidBoard	int, int	2D int[]	LOW_HINT
hasUniqueSolution	2D int[], int	boolean	INVALID_BOARD

6.2 Semantics

6.2.1 State Variables

N/A

6.2.2 Environmental Variables

N/A

6.2.3 Assumptions

N/A

6.2.4 Semantics of Exported Routines

- **generateRandomValidBoard(hints, uniquenessLikelihood)**

  Input:
  * hints: integer; number of hint digits to include in the output puzzle
  * uniquenessLikelihood: optional integer (default = 5); number of iterations to check for non-unique solutions. Higher values increase uniqueness of possible solutions.

  Output:
  * 2D 9x9 int list representing a newly generated Sudoku puzzle

  Exceptions:
  * LOW_HINT, if the number of hints < 24 which results in guaranteed non-uniqueness.

- **hasUniqueSolution(board, attempts)**

  Input:
  - board: 2D 9x9 integer list representing the cells on the game board
  - attempts: optional integer (default = 5); number of attempts to search for a different solution.

  Output:
  - True, if non-unique solutions are not discovered after specified number of attempts
  - False, if otherwise

  Exceptions:
  - INVALID_BOARD, if board is unsolvable

7 GUI Module

7.1 Syntax

7.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
constructor		HTML GUI	

7.2 Semantics

7.2.1 State Variables

N/A

7.2.2 Environmental Variables

Keyboard input from user
Image file upload from user
HTTP request metadata from user, including browser type, version, resolution

7.2.3 Assumptions

Back-end application has been started.
User is connecting through a major modern browser (Chrome, Edge, Firefox, Safari).

7.2.4 Semantics of Exported Routines

N/A

8 Flask Main Module

8.1 Syntax

8.1.1 Exported Routines

Name	Inputs	Outputs	Exceptions
__init__	string	string	ERR_MODEL
route	HTTP GET request	HTML	ERR_ROUTING

8.2 Semantics

8.2.1 State Variables

app: Flask app instance
cv: SudokuCV image recognition instance

8.2.2 Environmental Variables

cvModel: Keras classification model file used for image recognition
home, upload, recognize, solution, play, instructions: HTML template files for HTTP response rendering

8.2.3 Assumptions

Hosting environment is capable of running the Flask application.

8.2.4 Semantics of Exported Routines

- **__init__(modelFile)**
  Input:
  - modelFile: Keras classification model file path

  Transition:
  - An instance of SudokuCV is loaded into cv using modelFile
  - An instance of Flask is created and referenced in app

  Exceptions:
  - ERR_MODEL, if specified model file path is invalid or the file cannot be loaded
- **route(url)**
  Input:
  - url: the routing path from the HTTP request

  Output:
  - url = "/" renders home HTML
  - url = "/upload" renders upload HTML
  - url = "/recognize" renders recognize HTML
  - url = "/solver" renders solutions HTML
  - url = "/play" renders play HTML
  - url = "/instructions" renders instructions HTML

  Exceptions:
  - ERR_ROUTING, if url is not a valid routing address

9 Major Revision History

March 15, 2022 - Rough draft of sections
March 17, 2022 - Added MIS for SudokuCV, CVError, and CVResult
March 18, 2022 - Revision 0 complete