Automated Sudoku Solver

Shirish Waghe, Nayan Gadhari, Srijit Acharya, Dr. Rohini Palve

Dept. of Computer Engineering, Terna Engineering College, Nerul, Navi Mumbai, India

---***---

***Abstract* -** The popular Japanese puzzle game Sudoku is one of the most popular puzzle games of all times which is based on logical placement of numbers. The prime objective of this review is to do literature survey of methods to recognize Sudoku puzzles containing numerical digits from images taken with a mobile camera and puzzle solving techniques. An image in general portrays a visual perception of an artifact, for example a photo or a two-dimensional picture depicting any object, place or a person etc. In Computer Science, a digital image is a numeric representation of a two-dimensional picture, which often contain texts which are in a human readable format. These texts in a digital image are defined by a set of pixels. The paper discusses about various methods for digital detection and interpretation of a Sudoku puzzle using optical character recognition and vision-based techniques and solving the subsequent puzzles using various computer algorithms.

***Key Words*:** Computer Science, Image Processing, Fruit grading, visual perception, Image analysis.

1.INTRODUCTION

Sudoku is a single player, logic based, combinatorial, numeral digit placement puzzle. As a logic puzzle, Sudoku is an excellent brain game which at times can be highly addictive in a good way. It is stated that by playing Sudoku on a daily basis has improvements in one's concentration level and overall brain power. The puzzle basically consists of a 9x9 grid, constituting to form 81 cells in total . The main objective of this puzzle is to fill this 9x9 grid so that each row, each column and each of the nine 3x3 sub-grids that forms the grid accommodates all the digits from 1 to 9. A partially completed grid is provided by the puzzle setter, which for a precisely created puzzle has a single solution. Figure 1 shows a standard Sudoku puzzle. Sudoku puzzles normally appear in newspapers and other text and digital media with varying difficulty levels, but often their solutions are provided the next day. At the same time, learning to play Sudoku can be a bit intimidating for beginners. Our motivation for this project is to develop an Android application which can present an accurate solution to the user in quicktime.

- Sudoku is the Japanese abbreviation of a phrase meaning **"The digits must remain single"**.
- **Sudoku Contains:**

i. **Box (Region, Block):** A region is a 3x3 box. There are 9 regions in a traditional Sudoku puzzle.

ii. **Cell (Square):** Is used to define the minimum unit of the Sudoku board.

iii. **Candidates:** The number of possible values that can be placed into an empty square.

iv. **Clues:** The given numbers in the grid at the beginning.

Grid (board): The Sudoku board consists of a form of matrix.

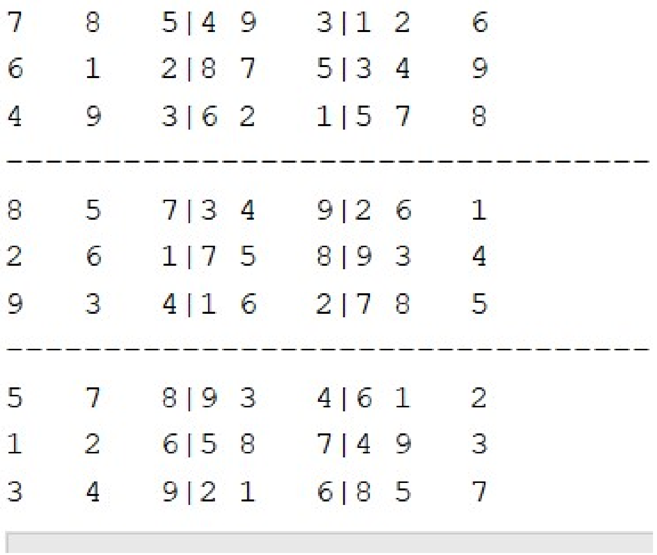
Sudoku solver using Image processing is a game solver. There isn't any direct application of this project. But the algorithms we will be building in this project have applications in Image Processing, Image & Digit recognitions. For eg.: These algorithms can be used for removing noise, sharping or brightening an image making it easier to identify key features. Solving Sudoku has been a challenging problem in the last decade. The purpose is to develop more effective algorithm by scanning and recognizing the clear image of Sudoku puzzle.

2. MATERIALS AND METHODS

A. Preprocessing:

Initially, we are verifying sudoku puzzle from the user. This image is provided to user to take input of puzzle as shown in fig.1 which then recognize the position of the numbers in the image.

After recognizing the numbers in the puzzle, the recognized number is then pass to parser to solve sudoku. The image of parser given below in fig.2:



```
7   8   5|4  9   3|1  2   6
6   1   2|8  7   5|3  4   9
4   9   3|6  2   1|5  7   8
---------------------------------
8   5   7|3  4   9|2  6   1
2   6   1|7  5   8|9  3   4
9   3   4|1  6   2|7  8   5
---------------------------------
5   7   8|9  3   4|6  1   2
1   2   6|5  8   7|4  9   3
3   4   9|2  1   6|8  5   7
```

Fig.1: Parser

B. Implementation:

Sudoku puzzles available in newspapers, magazines, etc is a small game of numbers where we have to write all numbers from 1 to 9 without repeating that number in a row, column and in small grid. Image will be captured of that sudoku and will be scanned. Complete Image will be scanned by using (Opencv) and after applying various image processing techniques data of that sudoku will be extracted. Extracted data will be sent to the xml parser where the numbers present in particular grid will be stored in xml format. Data stored in xml will be sent to the sudoku solving algorithm where the algorithm will solve the Sudoku and give the solved sudoku as output. The solved sudoku will be displayed in a new window. As shown in Fg.3: is the flow chart for Enhancing the image.

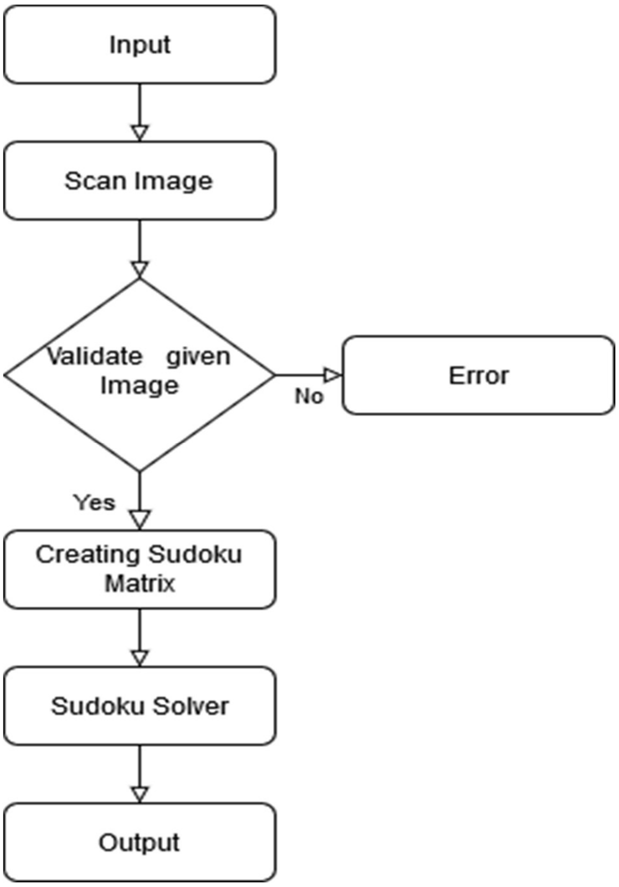


Fig.3: Representing flowchart working of website

4. MODEL EVALUATION

4.1 Evaluation Metrics

Following are the evaluation metrics used for this project:

1. Does the model detect the numbers properly?
2. How much time does it take to detect numbers?
3. How precisely the numbers are being detected?

4.2 Experimental Setup

For this project, the machine had various software's install such as:

1. Python3
2. PyCharm
3. Python IDLE

Various Python modules were also used:

1. OpenCV
2. Numpy
3. Pandas

4.2.1 Description of Data

The data that has been used in this project comprises of various mango images which are labelled using the LabelImg software to indicate the position of the mango and quality of the mango. The labelled data is collected into a csv file which is converted into a records file. This file is used to train the CNN (Convolutional Neural Network) model.

4.2.2 Methodology Used to Perform Experiment

The implementation of the project can be divided into three parts:

4.2.2.1 WebApp

Cross-Platform App Development Platform:

To develop an application that can serve both the Android and iOS in the limited amount of time, we chose to use Flutter. Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Since, Flutter is directly compiled to ARM code, it is a lot faster than React Native, on which the current Instagram app is based, in which the existing code is converted to native code and then run.

BLoC:

The code architecture follows the BLoC pattern prescribed by Google, the company behind Flutter. The

Business Logic Component (BLoC) pattern is a pattern created by Google and announced at Google I/O

'18. The BLoC pattern uses Reactive Programming to handle the flow of data within an app.

A BLoC stands as a middleman between a source of data in our app (e.g an API response) and widgets that need the data. It receives streams of events/data from the source, handles any required business logic and publishes streams of data changes to widgets that are interested in them.

A BLoC has two simple components: Sinks and Streams, both of which are provided by a StreamController. We add streams of event/data input into a Sink and listen to them as streams of data output through a Stream.

A StreamController can be accessed via the 'dart: async' library or as a PublishSubject, ReplaySubject or BehaviourSubject via the rxdart package.

4.2.2.2 Backend

We used Django, which is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Django is fast, secure, scalable and versatile. The existing Instagram application also uses Django for its backend. The following are the advantages of Django:

Object Relational Mapping:
The object-relational mapping layer (ORM) that can be used to interact with application data from various relational databases such as SQLite, PostgreSQL and MySQL. The Django ORM is an implementation of the object-relational mapping (ORM) concept which is a programming technique for converting data between incompatible type systems using object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.

PostgreSQL:
To have better integration with Django and a completely open source project we chose to use PostgreSQL. PostgreSQL is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. PostgreSQL is ACID-compliant, transactional, has updatable and materialized views, triggers, and foreign keys. It also supports functions and stored procedures.

PostgreSQL uses tables, constraints, triggers, roles, stored procedures and views as the core components that you work with. A table consists of rows, and each row contains a same set of columns. PostgreSQL uses primary keys to uniquely identify each row (a.k.a. record) in a table, and foreign keys to assure the referential integrity between two related tables.

Sprinkle Design Pattern:
While using the Django ORM, each type of Object has its own set of managers that retrieve a particular queryset. This is an effective way of having clean backend processing of requests.

4.2.2.3 WebApp Interface

To establish communication between the Frontend and the Backend, Django Rest Framework, a toolkit for API development is used. It allows for the following perks:

Django Integration:
The Django Rest Framework, was designed for optimal Django integration and can communicate smoothly with a Django backend, making request processing faster.

Token Authentication:
Authentication is the mechanism of associating an incoming request with a set of identifying credentials, such as the user the request came from, or the token that it was signed with. The token authentication works by exchanging username and password for a token that will be used in all subsequent requests so to identify the user on the server side. This means that the user's username and password don't need to be stored, nor do they have to login repeatedly. Whenever the user logs out from any of its devices the token is invalidated and old token cannot be used for logging in, this adds up as a security feature.

5. RESULT & DISCUSSION

Through generating this Sudoku solver, we have improved our programming ability. This was perhaps the largest program in terms of time invested and lines of code written that we have created. We have experienced participating as a team in what could be called a small research project.

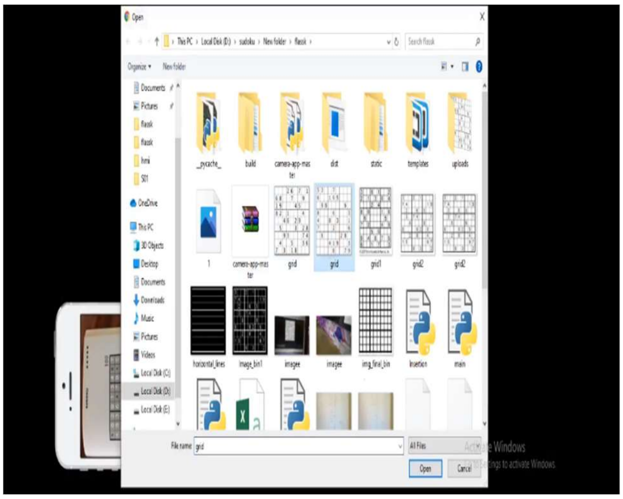
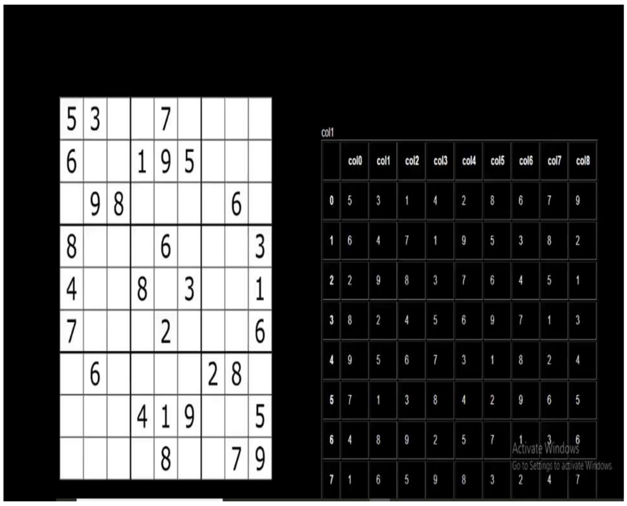


Fig.: 4.1: Selectin image from desktop



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

col1

	col0	col1	col2	col3	col4	col5	col6	col7	col8
0	5	3	1	4	2	8	6	7	9
1	6	4	7	1	9	5	3	8	2
2	2	9	8	3	7	6	4	5	1
3	8	2	4	5	6	9	7	1	3
4	9	5	6	7	3	1	8	2	4
5	7	1	3	8	4	2	9	6	5
6	4	8	9	2	5	7	1	3	6
7	1	6	5	9	8	3	2	4	7

Fig.4.2: Output

6. CONCLUSIONS

This project describes the working of a object detection model from training to implementing the trained model to detect object, in our case number, from user.

Thus, to train and implement a object detection model we initially need a data-set of the numbers to be detected which will be consisting images of this numbers. Then this image needs to be label to specify where the particular object is located. After a sufficient amount of time is passed while training the model it can be used to implement to detect various objects.

REFERENCES

1. Pramod J Simha K V Suraj Tejas Ahobala. “Recognition of numbers and position using image processing techniques for solving Sudoku Puzzles”. In: Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on, Mar. 2012.
2. Marlos C. Machadao Luiz Chaimowicz. “Combining Metaheuristics and CSP Algorithms to solve Sudoku”. In: Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on, Nov. 2011.
3. Akash Dutta Arunabha Ghosh. “Development of a Character Recognition Software to solve a Sudoku Puzzle”. In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Oct. 2016, 1–5. DOI: 10.1109/IEMCON.2016.7746076.
4. Snigdha Kamal Simarpreet Singh Chawla Nidhi Goel. “Detection of Sudoku Puzzle using Image Processing and Solving by Backtracking, Simulated Annealing and Genetic Algorithms: A Comparative Analysis”. In: Image Information Processing (ICIIP), 2015 Third International Conference, Dec. 2015. DOI: 10.1109/ICIIP.2015.7414762.
5. Baptiste Wicht Jean Henneberty. “Mixed handwritten and printed digit recognition in Sudoku with Convolutional Deep Belief Network”. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, Aug. 2015. DOI: 10.1109/ICDAR.2015.7333884.
6. Paritosh Pujari , Saurabh Thorat , Suraj Thopate , Yash Shah , Prasad Kharade ” Sudoku Puzzle Detection using Image Processing Techniques and Solving: A Literature Survey” In :- International Journal of Innovative Research in Computer and Communication Engineering Vol. 5, Issue 12, December 2017
7. PATRIK BERGGREN, DAVID NILSSON “A study of Sudoku solving algorithms” In: Bachelor’s Thesis at CSC Course: Degree Project in Computer Science, First Level DD143X