AUTOMATED PUZZLE SOLVER USING IMAGE PROCESSING

Project report submitted in fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Himanshu Malhotra (131219)

Hitesh Jain (131224)

Under the supervision of

Dr. Pradeep Kumar Gupta

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

Candidates' Declaration

We hereby declare that the work presented in this report entitled "**Automated Puzzle Solver Using Image Processing**" in fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science and Engineering**,** Jaypee University of Information Technology, Waknaghat is a genuine record of our own work carried out over a period from August 2016 to May 2017 under the supervision of **Dr. Pradeep Kumar Gupta,** Assistant Professor (Senior Grade) in **Computer Science and Engineering** department.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Himanshu Malhotra (131219)

Hitesh Jain (131224)

This is to certify that the above statements made by the candidates are true to the best of my knowledge.

Supervisor Name: Dr. Pradeep Kumar Gupta

Designation: Assistant Professor (Senior Grade)

Department name: Computer Science and Engineering

Dated:

ACKNOWLEDGEMENT

It gives us immense pleasure to present our knowledge-rich project, thanks for the continuous and patronage of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide **Dr. Pradeep Kumar Gupta** who made the way for us to move ahead through his suggestions and knowledge.

We also thank our friends who have helped us in successful completion of the project.

Date: Himanshu Malhotra (131219)

Hitesh Jain (131224)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviation	**Word**
OCR	Optical Character Recognition
PNG	Portable Network Graphics
JPEG	Joint Photographic Experts Group
GIF	Graphics Interchange Format
DLX	Dancing Links Algorithm

LIST OF FIGURES

ABSTRACT

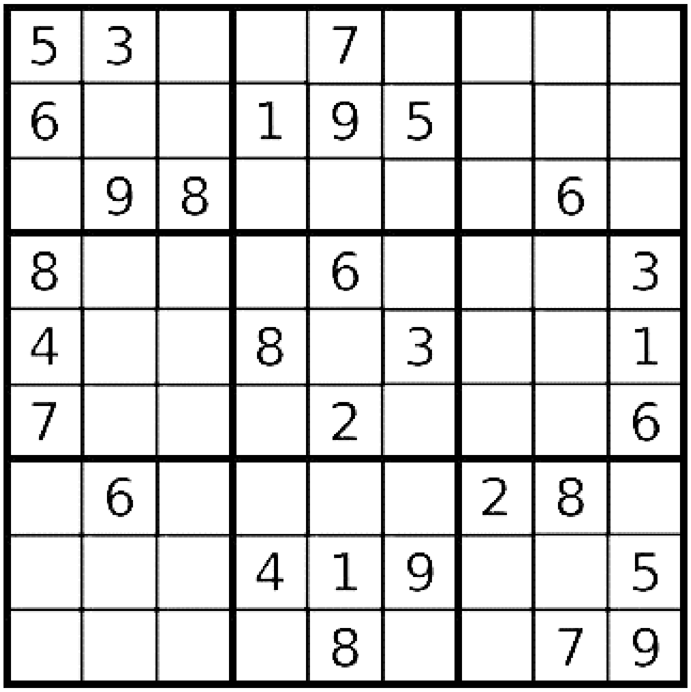
In our project we proposed a method of detecting and recognizing the elements of a Sudoku Puzzle and providing a digital copy of the solution for it. The method involves a vision-based Sudoku solver. The solver is capable of solving a Sudoku directly from an image captured from any digital camera. After applying appropriate pre-processing to the acquired image we use efficient area calculation techniques to recognize the enclosing box of the puzzle. A virtual grid is then created to identify the digit positions. Tesseract OCR (Optical Character Recognition) engine is used as a method for digit recognition. The actual solution is computed using a backtracking algorithm. Experiments conducted on various types of Sudoku questions demonstrate the efficiency and robustness of our proposed approaches in real-world scenarios. The algorithm is found to be capable of handling cases of translation, perspective, illumination gradient, scaling, and background clutter.

Chapter – 1

INTRODUCTION

1.1 Introduction

In real life, we run over Sudoku riddles of fluctuating trouble levels in daily papers and magazines. It is a typical relaxation action for many people. Be that as it may, it is watched that the arrangement and the solution is not generally promptly accessible for the user's verification. By and large, individuals need to hold up the Sudoku puzzle till the following day to check the arrangements of the Sudoku. Henceforth our inspiration for this project was to build up an application on an android gadget for this reason. In our application, the client needs to catch a perfect picture of an unsolved Sudoku Puzzle, which then returns the total arrangement of the same.



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

Figure 1.1: Example of Sudoku puzzle

Sudoku is a justification based game with the target to complete a 9x9 cross section so that each line, each portion and each of the nine 3x3 boxes contain the numbers 1 through 9 with a unique number. The issue itself is a pervasive brainteaser yet can without a doubt be used as an algorithmic task, with similarities to the graph shading issue just with settled size and conditions. The present day type of Sudoku was introduced in Japan in the eighties and has starting now and into the foreseeable future pulled in an impressive measure of programming architects and thusly there are a considerable measure of different Sudoku fathoming counts open on the Internet. Sudoku solvers are captivating because they are sensible employments of incredibly speculative issues, and a large number individuals think about them.

1.2 Problem Statement

The assignment is to tackle the Sudoku grid in as less time as possible. It should be possible by researching diverse methods for Sudoku solving and contrasting them for the most productive arrangement. Sudoku itself can be tackled utilizing brute-force approach in a sensible measure of time as a rule, yet there are extraordinary situations where it requires a long investment to the brute-force method. In this manner our idea is to attempt to discover efficient calculations for all examples of the issue and assess them while utilizing the ideal solver to get the answer for the Sudoku puzzle.

There are two primary limitations that decide the productivity of an ideal solver. They are Time utilization and Memory utilization whose level of fulfilment decides the nature of an ideal Sudoku solver. These imperatives as a rule fluctuate from calculation to calculation.

To understand the Sudoku puzzle in however much less time as could be expected. It should be possible by exploring distinctive strategies for understanding Sudoku and looking at them for the most productive arrangement. Along these lines our assignment is to attempt to discover effective calculations for all occasions of the issue and assess them while utilizing the ideal solver to get the answer for the Sudoku grid in as less time as possible and taking as less memory as possible.

1.3 Objective

The goal of this project is to click a picture from a mobile phone and send it to the server via Wi-Fi and then using that image we will explain Sudoku with the utilization of recursive backtracking algorithm while following the rules and then the generated Sudoku image needs to be transferred back in the mobile phone via Wi-Fi

The Constraints are as per the following:

- Sudoku puzzle can just contain the numbers 1 through 9. A position requirement: Only 1 number can possess a cell, otherwise the output will be wrong and it is an unsolvable Sudoku
- A push imperative: Only 1 example of a number can be in the column
- A segment requirement: Only 1 occasion of a number can be in a section
- An area requirement: Only 1 occasion of a number can be in an area

1.4 Methodology

The process to tackle the Sudoku depends on a recursive backtracking methodology. We store the numbers taken from the grid in the picture, in a two dimensional matrix with the number 0 allocated to clear areas which doesn't have any digit associated to them. To get an answer for the Sudoku grid, we recognize a clear area in the matrix. In light of Sudoku principles, we recognize a very tough task by repeating through the numbers 1-9. We then attempt to recursively show the matrix with this new number position. In the event that there are no more grid areas which should be filled, it means that the network is unsolvable, and we give back the unsolved matrix back to the fundamental calculations. Then again, on the off chance that we neglect to recursively discover an answer for a matrix, we backtrack and attempt an alternate number task for an area and take a chance at settling the network once more. On the off chance that all number assignments (1 to 9) are exhausted without finding an answer, we finish up with the conclusion that the matrix is unsolvable and in this way the solver will return false. For our motivations anyway, we accept that the picture taken by the client is a Sudoku grid and our calculation will dependably explain it regardless of the trouble level of the puzzle.

The strategy for illuminating the method is partitioned into:

- Capturing a picture of Sudoku utilizing camera
- Sending it to the server for processing via Wi-Fi

- Smoothing the picture and pre-preparing methods
- Grid discovery
- Grid Extraction
- Recognition of digits utilizing OCR
- Using different calculations to get the answer for Sudoku
- Comparison between various calculations utilized for tackling Sudoku Environment setup for Android usage
- Building the required capacities and class layouts for the application
- Generate the desired image of solved Sudoku grid
- Send the image back to the mobile phone via Wi-Fi

An android powered mobile phone and a Sudoku grid image is required for the whole process.

CHAPTER – 2

LITERATURE SURVEY

Edge Detection:

In [1] developers have proposed a way involving computations to deal with edge location. The accomplishment of the approach depends upon the meaning of tough arrangements of objectives for the calculation of edges. These objectives must be sufficient enough to restrict the conduct of the identifier while making negligible joining about the type of the arrangement. We characterize discovery and restriction criteria or a class of edges, and present scientific structures for these criteria on the administrator. A third model is then added to guarantee that the locator has just a single reaction to an edge. We utilize the criteria in numerical enhancement to know finders for a few basic picture highlights, including step edges. On practicing the examination to step edges, we find that there is a characteristic vulnerability guideline amongst identification and limitation, which are the two fundamental objectives. With this guideline we came to know an administrator shape which is ideal at any scale. The ideal indicator has a straightforward rough usage in which edges are set apart at maxima in angle extent of a Gaussian-smoothed picture. Gaussian blur is to be applied for retrieval of edges .We expand this straightforward indicator utilizing a few widths to adapt to various flag proportions in the picture. We show a general technique, called highlight, for the fine joining of data from administrators at various scales. At last we demonstrate that progression edge finder execution which enhances significantly as the administrator point spread capacity is stretched out along the edge. This location helps in utilizing a few lengthened administrators at every point, and the directional administrator yields are incorporated with the slope most extreme indicator. They have depicted a technique for the plan of edge locators for discretionary edge profiles. The plan depended on the detail of location and limitation criteria in a frame. It was important to increase the first two criteria with a good reaction measure keeping in mind the end goal to completely catch the instinct of good discovery. A frame for the criteria was introduced, and numerical advancement was utilized to discover ideal administrators for all the edges. The experiment was then confirmed to thought of ideal administrators for step edges. The outcome was a class of administrators related by spatial scaling. There were immediate trade-offs in discovery execution versus confinement, and this was controlled by the spatial width. The motivation reaction of the ideal edge administrator was appeared to summarize the main subsidiary of a Gaussian. An identifier was proposed which utilized thresholding with hysteresis to take out variation of edge shapes. The edges were set by measure in the picture, as controlled by an estimation. This identifier made utilization of a few widths to adapt to fluctuating picture flag to common proportions, and administrator yields were joined utilizing a technique called union, where the reactions of the little administrators were utilized to anticipate the extensive administrator reactions. On the off chance that the real expansive administrator yields contrast essentially from the anticipated qualities, new edge focuses are correct. It is in this way to depict edges that happen at various scales, regardless of the possibility that they are spatially incidental, we were able to detect the edges. In two measurements it was demonstrated that stamping edge focuses at maxima of slope size in the inclination which is identical to discovering zero-intersections of a specific nonlinear differential administrator. It was demonstrated that when edge forms are locally straight, the directional administrators will give preferred outcomes over administrators with a round support. A strategy was proposed for the productive era of exceeding styles at a few introductions. Among the augmentations of the work, the most fascinating unsolved issue is the combination of various edge identifier yields into a solitary portrayal. A plan which consolidated the edge and edge finder yields utilizing highlight blend was executed, however the outcomes were uncertain. The issue is significantly more confused here than for edge administrators at various scales on the grounds that there is no unmistakable motivation to lean toward one edge sort over another. Every edge set must be blended from the other, without an inclination brought about by overestimation in one course as created by overestimation in one heading. The criteria they have given can be utilized slight change for the outline of different sorts of administrator. This is how we retrieve the edges by applying Gaussian Blur and adaptive thresholding to retrieve the Sudoku grid as expected.

Continuous image capture using android device

Yixin Wang [2] made An Android application which permits a Sudoku puzzle to be solved and explained continuously utilizing caught pictures from the phone’s camera. The client sees a close continuous encouragement of the phone’s camera. In the event that the camera sees a Sudoku grid with sensible clarity, the puzzle will be illuminated and the registered qualities will be overlaid on top of the camera with the correct point of view. While the majority of the means of the puzzle in this process are fundamentally the same as existing Sudoku solvers, the calculation contrasts in how the cell areas are resolved. Normally, the Hough transformation which we are also applying in this puzzle solver as we need it very much is utilized to distinguish the framework lines which are then used to segment the puzzle into cells. In any case, this is not strong to the situation where the framework lines don't seem straight in the picture, which happens when the paper containing the reason is not at the same level. In this process or project, every matrix cell is found to be exclusively utilizing and organizing components to better handle situations where the lines and segments of cells are not very much straight. A calculation for consequently tackling Sudoku puzzles from versatile camera pictures were depicted. While large portions of the means of the calculation are fundamentally the same as existing Sudoku solvers, the matrix wrapping step maintained a strategic distance from a notwithstanding portion for utilizing and organizing components to access every cell one by one or separately. This change diminishes the likelihood that a digit is separated from positioning or being mistakenly allocated to an alternate cell than the one it really is in. The last application could effectively separate digits from non-planar riddles and riddles saw from a point.

Optimal approach to solve a Sudoku Puzzle

In [3] developers have proposed a calculation which is smaller than normal network based, i.e., we need to separately experience nine little matrices (rather than 81 cells) and do backtracking just on them, which is less tedious. In addition, no speculating is included in the entire calculation. In this paper an approach has been made to build up a calculation which is smaller than normal matrix based, i.e., we need to separately experience nine little frameworks (rather than 81 cells) and do backtracking just on them, which is a bit of an easy task to perform . Also, no speculating is included and no excess calculation is performed in the entire calculation. In this paper they have proposed a restrictive smaller than normal matrix based Sudoku solver calculation, which is totally optimal and effective and very fast. The solver considers each of the scaled down networks (rather than clear cells in detachment) of size 3×3 every cell that has been created without previous element for planning such a calculation for a given Sudoku grid of size 9×9. In our calculation a pre-handling is there for figuring just all the required changes for each of the smaller than usual matrices in light of the pieces of information in the given Sudoku puzzle. It has been seen in the extremely worst case circumstances that the quantity of substantial stages is considerably not exactly the aggregate number of needed changes for each of the smaller than normal frameworks, and regardless of the possibility that there are less hints in a few scaled down matrix of an occasion, it introduced four contiguous line and section with little lattices which seriously helps in decreasing a definitive number of needed changes for that little network as well. Anyways, this approach of small framework and the calculation of legitimate stages and checking their similarity among line with little networks and section are scaled down to matrices which are completely new and accomplished without this area of work. As they consider smaller than usual matrices for finding just the substantial arrangements of a given Sudoku puzzle as opposed to considering the individual (clear) cells, in this manner, the calculations required in the calculation is altogether decreased or reduced to a very big amount which was the goal and now the goal is achieved. On an account of a 9×9 Sudoku puzzle, there are 81 such cells with a very less and very small pieces of information (which is less) and the staying clear cells (which is more than enough), while there are just nine such small scale frameworks each of which comprises of 3×3 cells. Here, the perception to the Sudoku is not via seeking of missing numbers cell by cell (as though looking location by traveling through all possible combinations in the worst case); rather, it is one stage over the ground of the puzzle by considering gatherings of cells or smaller than usual lattices (and seeking the same from a superior view). The brilliant and awesome thing about the calculation created in this paper is that a similar structure can likewise be straight away connected for bigger Sudoku examples. For example, 16×16, 25×25, 36x36 etc. or of whatever other rectangular sizes (with their separate target capacities). The level of trouble is another critical issue that all the prior Sudoku solvers consider while building up a calculation. There are no firm which decides that the factor expressing the trouble level of a Sudoku puzzle. A filled Sudoku puzzle might be greatly simple to understand, while a thickly filled Sudoku grid may really be harder and tougher to split. From a programming perspective, we can decide the trouble level of a Sudoku grid by the amount that how much effort must be used to explain the puzzle, and the diverse levels of trouble are simple, medium, troublesome, to a great degree troublesome, and so on. By chance, the Sudoku solver created in this paper does not explain any level of trouble; rather, all the Sudoku occasions are having a similar level of trouble. Now and again, more substantial changes might be produced for a few scaled down area, yet all in all, the quantity of stages is a great deal and the little matrices with less or maybe very less in number of legitimate changes in the long run for all the correct answers for a given Sudoku case.

In [4] the author proposes a strategy or a way for separating a Sudoku puzzle caught in a picture. Artificial Intelligence methods can then be connected to illuminate or sense the Sudoku puzzle. A particular engineering is made for this reason. Modules can be supplied as required, making it simpler to enhance and keep up an application utilizing the purpose of engineering required to solve a Sudoku puzzle. This paper has elements which are utilized to identify and separate the Sudoku grid from a picture. A scheduled design for this reason for existing is proposed. Since different strategies are adopted as a part of the way towards removing the Sudoku puzzle, a scheduled engineering takes into account change and practicality of an execution of this procedure as individual module usage can be changed all the more effortlessly without influencing whatever is left of the product execution. Picture preparing parts give us the advanced matrix of the Sudoku grid. This has been put away as a fitting information structure and after that given to an Artificial Intelligence module for traversing the puzzle. Like the picture preparing steps, the Artificial Intelligence segment is scheduled. Thus, it is anything but difficult to swap out the calculation utilized for improving the Sudoku grid or enhance the Artificial Intelligence execution bit by bit, after some time. Separating the Sudoku puzzle from a picture includes the Gaussian blur and thresholding all in all:

1. Pre-handling the picture
2. After the pre-handling, retrieval of Sudoku is done
3. Finding the Sudoku matrix's external box
4. Finding the corners of the container
5. Utilizing the container corners to concentrate cells
6. Recognizing the number present in every cell
7. Making a virtual Sudoku network

8. Solving the Sudoku using the recursive algorithm (Backtracking)

Different picture handling and PC vision systems might be connected at every individual’s system. Pre-handling contributes the resizing of picture to a fitting size, Gaussian blur and thresholding. The benefit of taking after this engineering is its clarity or execution of the Sudoku solver, if a module's usage needs to be enhanced or improved, it can be changed effortlessly, the length of the contribution to the module and yield from the module stays unaltered. Such a structural model would be valuable for creating applications which chip away at Sudoku difficulties, as the measured nature considers rapidly altering the way of the desired output provided by essentially changing parameters or modules of the design.

Filtering Technique to solve Sudoku

In [5] creators highlight the present ease of use issues when tackling Sudoku issues. This issue is an outstanding puzzle diversion which comprises in ruling out or eliminating numbers in a board, usually of 9 × 9 grid. The leading body of the amusement is made out of 9 segments, 9 columns and 9 (3 × 3) sub-lattices; every one containing 9 cells with particular whole numbers from 1 to 9. A puzzle is finished when all cells have ruled out, and the past limitations are fulfilled. A few examples are exceptionally hard to understand, to handle this issue, we have utilized a separating and distinguishing system named Arc Consistency 4 (AC4) from the Constraint Programming areas. This calculation is uncovered which is tremendously identified with the procedures utilized by clients keeping in mind the end goal to understand the Sudoku occasions, yet conversely, this strategy is executed in a brief timeframe, offering a decent determination manual for the clients. By and large, separating strategies make less understanding Sudoku puzzles, giving great data to clients. The Arc Consistency property originates from a Constraint Programming (CP) space. An issue which is expressed as far as factors and requirements is called Constraint Satisfaction Problem (CSP). Constraint Programming (CP) is a programming worldview where the numerical issues are expressed as factors and afterwards comprehended by a Constraint Programming (CP) solver. Keeping in mind the lasting goal to determine a Constraint Satisfaction Problem (CSP), the conditions of the variables are sought and the limitations must be fulfilled all the while. This property builds up a connection between two factors and keep up just attainable values in every variable areas, allowing to diminish the search space, erasing the unfeasible qualities. To guarantee that an issue is curve faster and efficiently, it is expected to apply a channel procedure, existing difficult calculations to guarantee the property of solving, the calculation utilized as a part of this examination was Arc Consistency Algorithm #3, characterized by Alan Mackworth. In this paper work, they have displayed an extremely helpful system with a specific end goal to encourage the determination procedure of the Sudoku grid. This diversion is known to be a NP-Complete (Not polynomial time) issue, being a difficulty in research fields like Constraint Programming and Meta heuristics, including cases extremely hard to calculate. They have utilized the AC4 calculation for this as a part of request to offer a determination manual for the clients. This system is like the methodologies utilized by clients at the determination procedure, however this awesome and spectacular strategy is executed in a brief span, making less demanding and quicker the procedure of the Sudoku puzzle and the grids of 3x3 or 9x9. An example of the reverting is finished, when all cells have a null value that means there is no value in that particular cell, however it is not all that simple, and the digit must go along the issue imperatives. The limitations can be lessened to a articulation which says, each cell must have a number, it must be one of a kind in the line, segment and sub-matrix which it has a place with the Sudoku grid. At that point, if the announcement is done in single limitations, we have 3 requirements for each cell making this task exceptionally hard to illuminate. Such a reason or cause has a place with the NP-Complete accumulation of issues, this issue has broadly been considered for proposing a few methodologies for the most recent years to understand it, from finish seek techniques, for example, requirement programming and Boolean satisfiability and dynamic programming to fragmented hunt strategies, for example, hereditary programming, meta heuristics as a rule and mixture calculations. The fundamental enthusiasm of the Sudoku issue is because of a few occasions which are exceptionally hard to settle. So we need to handle this issue as it is not needed at this moment. To handle this issue, we have utilized a good procedure as a part of request to approve the request from clients, also to encourage the Sudoku determination and figuring out if or not is an answer in a prior phase of the determination procedure by clients, in connection to allot and given qualities. The separating procedure picked is named Arc Consistency 4 (AC4) from the Constraint Programming area, which is a calculation that authorizes the curve consistency property on a limitation arrange. In the wake of evaluating its determination procedure, we have understood that the determination stage is highly identified with the methodologies utilized by clients to understand the examples, however conversely, this strategy is executed in a brief span, offering a decent determination manual for the clients. The viability of separating strategies was tended to playing out an assessment; it was acknowledged with a specific end goal to distinguish the nature of sifting over the diverse challenges of the Sudoku puzzle. When all is said in done, separating methods make less demanding and simpler Sudoku puzzles, giving great data to clients to this. The benefit of taking after this engineering is its clarity or execution of the Sudoku solver, if a module's usage needs be enhanced or improved, it can be changed effortlessly, the length of the contribution to the module and yield from the module stays unaltered. Such a structural model would be valuable for creating applications which chip away at Sudoku difficulties, as the measured nature considers rapidly altering the way of the desired output provided by essentially changing parameters or modules of the design.

Development of character recognition software

In [6] the have strived to build up a framework which can remove writings from a computerized picture of a Sudoku puzzle, illuminate the puzzle, and after that give an answer to the puzzle. Our approach is particular yet its application are changed. So in this, firstly the camera captures the image and then that image is sent via a device and then server receives the image and do the actual processing to process or retrieve the Sudoku puzzle or 9x9 grid from the image, then Gaussian blur and methods like adaptive thresholding are applied to retrieve the grid and then they break the grid into 3x3 different grids, they are 9 in number. And then we try recursive algorithm to solve the Sudoku, if the image is fine and solvable we will solve the Sudoku and generate an image and send back to the user or Mobile Phone.

SOFTWARE REQUIREMENTS:

1. Python: It is server side scripting language, easy to code and having simple and meaningful syntax
2. Open CV (Open Source Computer Vision): It is a library of programming capacities fundamentally gone for computer vision, which is the field of handling and breaking down computerized pictures and make them understandable to the computer

Their work depends on a testing framework. We contrast each removed picture scrap and our effectively settled database and afterward assign piece to a number content. Development of this database and conforming size of given computerized picture frames the centre segments of the Pre-preparing phase of our work. At the very onset we have chosen 25 distinctive text style styles to speak to every one of the digits from 1 to 9. A progression of 25 pictures are then made with each containing the digits from 1 to 9 with every picture speaking to an alternate textual style and appearance. This is done to guarantee that we can outline sort of picture text style to its right individual digit. Their next goal is to get each of the 25 pictures we have made and after that concentrate the picture bit of every digit from every picture. The pixel estimations of these bits are put away with a label which speaks to the number which is shown in the picture piece. In this manner every picture has a comparing content record which stores the pixel estimations of the considerable number of digits from 1-9. In this way 25 content documents are framed comparing to the 25 pictures we had made before. Basically we have now put away the 25 pictures both in .png format and .txt design. We then continue to shape our primary database which is a gathering of all the 25 content records that we have made. Subsequently now we have 25 images of pixel data that indicate a digit. This enhances the exactness of our application. Hough Transform is a well-known procedure to identify any shape, in the event that you can speak to that shape in numerical frame. It can identify the shape regardless of the possibility that it is broken or distorted or any disturbance in the image. A line in its parametric frame is spoken to $\rho = x\cos\theta + y\sin\theta$ where ρ is the opposite separation from source to the line, and θ is the point shaped by this opposite line and level hub measured in counter-clockwise. Picture is resized by of Hough lines. At that point we continue to assign every given digit as a different picture piece which is then resized into a 16x16 picture and added to a rundown. It is to be noticed that the clear spaces are additionally added to this rundown as clear pictures. K Nearest Neighbour is one of the most straightforward order calculations for directed learning. The idea is relevant where we have to scan for a best match of test information in highlight space. Here K is a client characterized steady whose esteem relies upon the type of information. High estimations of K helps in removal of the impact of order yet makes separation between different classes less particular.

The calculation is separated into two unmistakable stages:

- Training Phase A: Master database is shaped which stores all the preparation vectors. The putting away of the considerable number of vectors and the naming of the different classes constitutes the preparation period of the calculation.
- Classification Phase: In this stage another unlabelled vector is presented and is then arranged as needed

Additionally in our calculation we have as of now made an ace database which contains pixel data of pictures (res: 16x16) of all digits from 1 to 9 in a quarter century text styles. This database is our preparation data. We realize that every picture is characterized by its 256 pixel qualities and therefore we accept 256-dimensional plane and place every one of the components of the preparation information. The estimations of the pixels at whatever point we have to analyze another picture piece it is likewise set in that plane and afterward K Nearest Neighbour calculation is connected to give us the best match in the element space. The best match is then mapped from a picture of a digit to a genuine digit and added to a content record. The significant downside of this calculation happens when the class appropriation is skewed that is substantial contrasts between class estimate which compels the calculation to arrange another inquiry into a huge class as it were. This issue, be that as it may, can be settled if the arrangements are weighted. A straightforward recursive calculation is connected in understanding a Sudoku confuse. Tolerating the disturbance of the puzzle we remove or rule out qualities which augment as we continue from cell by cell till we disregard any manage relating to a solitary cell or an empty cell which need to be filled by a number. When a violation of a number or anything like that happens we need to back track till the violation of image is settled by recursively try all the numbers from 1 to 9 in a cell. It is to be noticed that the calculation works just on a content document that has every one of the estimations of the puzzle in its initial nine lines and initial nine segments with no spaces. Every single clear space in the grid is allocated a pre-characterized estimation of 0, as we cannot assign 1 to 9 as these are the actual numbers needed to be fit in the Sudoku for the desired output, hence we use 0.

Adaptive document image binarization

In [7] author proposed another strategy displayed for versatile report picture binarization, where the page is considered as a gathering of subcomponents, for example, content, foundation and picture. The issues brought about by enlightenment and many source sort related corruptions. Two new calculations are connected to decide a nearby limit for every pixel. The execution assessment of the calculation uses test pictures with ground-truth, assessment measurements for binarization of printed and manufactured pictures, and a weight-based positioning methodology for the none (nil) result presentation. The proposed calculations were tried with pictures including diverse sorts of report parts and database managements. The outcomes were contrasted and various known procedures in the writing. The binarization procedure is meant to be utilized to organize in different report examination, preparing and recovery. In this way, the extraordinary record attributes, as printed properties, illustrations, line-drawings and complex blends of their design semantics ought to be incorporated into the prerequisites. Then again, the system ought to be basic while taking all the report examination requests into thought. Fig. 4 exhibits the general approach of the binarization preparing now. Since ordinary archive division and marking for substance investigation is out of question in this stage, we utilize a fast crossover switch that dispatches the little, determination adjusted windows to literary (1) and non-printed (2) edge assessment strategies. The change was produced to cover most bland appearances of run of the mill archive design sorts and can undoubtedly be altered for others too. The edge assessment methods are adjusted to printed properties, with the extraordinary resistance and recognition to various fundamental deformity sorts that are normally acquainted with pictures. The result of these strategies speaks to a limit proposed for every pixel, or each nth pixel, chosen by the client. These qualities are utilized to gather the final result of the binarization by a limit control module. The method likewise empowers the usage of multi-limits area by locale of universally, if wanted. 3. Versatile binarization. The archive picture contains obstruction (surface) that can be partitioned into uniform, separating and momentarily evolving. The surface contained in pictures and foundation can ordinarily be more tasteful to uniform or separating classes, while the content, line drawings, and so on have more transient properties by nature. Our approach is to examine the nearby record picture surface with a specific end goal to settle on the binarization strategy required (Fig. 4). Using this choice, and a mixture exchanging module chooses one of two particular binarization calculations to be connected to the area. The objective of the binarization calculations is to deliver an ideal edge esteem for every pixel. A quick alternative is to figure an edge for each nth pixel and afterward utilize addition for whatever is left of the pixels (Fig. 5). The binarization strategy can likewise be set to sidestep the half breed switch stage. At that point the client can pick which calculation is chosen for thresholding. Every single other module work in an indistinguishable route from in half and half conditions. The following section depicts the locale sort and exchanging calculations. The two obstruction binarization calculations are then talked about in detail. The final binarization is performed utilizing the proposed edge values. This procedure is portrayed in the last subsection. 3.1. Area investigation and exchanging Threshold calculation is gone before by the choice of the best possible binarization technique in view of an examination of nearby picture properties. In the first place, the record picture is tiled to parallel estimated rectangular windows of 10*20 pixels wide, relating to the determination that sprightly fluctuates somewhere around 75 and 300 dpi. Two basic elements are then registered for every window; these outcomes are utilized to choose the strategy. The rest highlight is basically the normal dim estimation of a window. The second element, transient measures neighbourhood changes interestingly. The distinction qualities are collected in every sub window and after that scaled somewhere around 0 and 1. Utilizing the breaking points of 5%, 20% and 30% of scaled qualities, the transient concession property is defined as uniform, near-uniform, deferring or transient. This course division is made by homogeneity at first glance. As indicated by these names, a vote is given to comparing binarization technique that will be utilized as a part of a window. These elements are very useful in solving the puzzle efficiently and faster with less number of errors. The names near-uniform and uniform can relate to foundation and scene pictures, and offer votes to the SDM. The marks deferring and transient give their votes to the TBM technique which calculates the amount better. Determination of a binarization calculation is then executed as taking after illustration rules (1, 2) appear.

There are 2 rules which are needed to follow to solve this issue, which are as follows:

1. In the event that the normal is medium and a worldwide histogram pinnacle is not in a similar quarter of the histogram and transient respect is uniform, then utilize TBM

2. On the off chance that the normal is high and a worldwide histogram is in a similar quarter of the histogram and transient concession is transient, then utilize SDM

The white areas are guided to the SDM calculation, while the dark districts are binarized with the TBM. Binarization of non-printed segments. As in delicate control applications, our calculation first investigates the window surface by figuring by making qualities clear and a bit more explainable. At that point, the delicate control calculation is connected to each nth pixel. The outcome is a nearby edge in view of neighbourhood locale attributes.

To guarantee nearby adaptively of edge determination, we utilize two unique sorts of privately computed highlights:

1. Histogram-based investigation plans and components are regularly utilized as a part of binarization techniques. In record investigation the histogram is extremely valuable for recognizing and separating areas in physical and legitimate examination
2. We utilize another approach created for nearby recognition and weighting of limits in dim scale picture surface. Another component called weighted bound (=b) is presented and used in the delicate control calculation

The weighted bound (b) is utilized for portrayal of neighbourhood pixel profiles by following low, medium and high pixels in a little zone. In a given surface region of N x N pixels, where n is a window width picked up from the non-covering areas examination tile estimate three distinct measures are taken into consideration. The qualities are gathered in a two-dimensional table used to weight and streamline the three envelope bends in delicate control enrolment capacities. The proposed binarization calculation was tried with the benchmarking strategy and different situations against a few known binarization procedures in the writing. Utilizing the earth elements, (for example, extraordinary corruptions) and accessible report and test picture databases the calculation results were assessed and benchmarked against each other, against the ground-truth learning by visual and benchmark event(s) assessment forms. The emphasis was determined to reports with printed content and on multi-content archives, i.e. records having content, design, line drawings and halftone. The test pictures were chosen from an uncommon database of archive picture classifications, involving more than 1000 ordered report pictures (e.g. article, letter, reminder, fax, diary, logical, outline, and so forth.) .The numerical test and results introduced were picked up utilizing binarization measurements accelerating the execution in printed picture area binarization. It exhibits a case benchmarking scene performed to a database of 15 printed record pictures with best quality having light. Visual outcomes to a specimen input picture having 20% of focused light imperfection, a case of a ground-truth picture eliminates the aftereffects of the proposed and correlation binarization calculations. The outcomes demonstrate good conduct of Sauvala's, Nicblack's and Eikvikl's calculations, when the breaking point is set to 75% execution, i.e. the farthest point where the OCR execution drop is under 15% utilizing OCR bundle. Bernsen endured of confused and noisy disturbance, that was acquainted with binarized result picture, while the Eikvikl's edge governed a part of the darkest zones have a place with protesting pixels. Parker's calculation adjusted ineffectively to even little changes in lumination, yet had adequate outcomes with moderately and clean and neat dark scale record pictures for better usage in this case. The visual tests performed for a manufactured test picture database depended on positioning as indicated by obstacle goals set for these sorts of pictures. The reason for the manufactured picture database is to permit visual investigation of the nature and conduct of the benchmarking and awesome strategy in a very important sort of circumstance like as we did in this project, e.g. in edge detection, the question arises regarding conservation, in changing or fluctuating edges, and so on. This is planned to help the effectiveness and goodness choice of very important calculation to much ecological conditions as far as versatility to changes, shape administration, save the conservation, same position of locale protection, As the information of dark scale pictures were artificially produced, an arrangement of ground truth pictures were created cantering in obstacle zones of edges for measuring the calculation execution and conduct for the project of Sudoku solver. In this way, the benchmark results are relevant enough on the determination of the ground-truth set utilized, i.e. the objective execution gather the calculation conduct, as it is the optimal approach used. The edge and foundation consistency was utilized as a weight criteria, where the Euclidean separation was utilized as a separation measure between the outcome and the ground-truth pixel maps demonstrates a circumstance, where the manufactured picture has consistent foundation from white to dark, and thin lines, whose dim scale esteem floats on the other way from the foundation. This is how we can detect the edges and grid lines in this automated puzzle solver project. The test assessment criteria was determined to separate and differentiate the lines of Sudoku solver taken from mobile phone from foundation and consistency of the foundation.

Classification of digits in multi-application OCR System

In [8] the author talked about extraordinary arrangement of PC vision research and study is devoted to the frameworks intended and to be used to recognize and break down PC printed records and human composed content. Optical Character Recognition (OCR) allows the way toward changing over pictures written by hand, typewritten, or printed content into an arrangement contained by machines with the end goal of altering, ordering, seeking and a lessening away size. In this paper we have combined the usefulness of Optical Character Recognition and have concentrated on its applications like Image Sudoku Solver, Car License Plate Detection and Recognition, document reader, PDF reader, Handwritten and Computer Printed Documents Recognition. This paper builds up an easy to use application for performing picture to content change like altering values of Sudoku. The created framework is sorted out as an arrangement of modules, each committed to a particular application. Auto License Plate Detection and Recognition framework that removes out the License plate precisely and create an important amount of acknowledgment of the characters in the License Plate. With the proposed strategy, we have possessed the capacity to accomplish what comes about with 95% exactness for the tried pictures. Picture Sudoku Solver is planned to work with Sudoku Puzzle pictures by extracting or taking or retrieving out the numbers, limits from them and after that understanding the puzzle. In this we can separate, receive and settle around 95% of Sudoku Puzzles being tried for the reason. Web based Handwriting Recognition finishes the ongoing acknowledgment of client's Handwriting from a Mouse or wireless mouse or a Laptop Touchpad. Optical Character Recognition (OCR) is the short type of Optical Character Recognition or Reader. People have an optical instrument through which we can see, envision, perceive and read things. Eye is an organ that responds to the light approaching from the scene and move about as a contribution to the thinking. Eye retina have photoreceptors: Rods (delicate to shading) and Cons (touchy to light brilliance) because of which the light flags are changed over into the electric flags and are then transmitted to the mind for translation by means of the optic nerve.

Through this entire procedure a human can read and comprehend the content that he/she sees. Perusing implies visual handling of words. OCR is only a method to show or prove human perusing ability to machines. By utilizing the OCR system the machines can read, dissect and get it. The Optical Character Reader Application includes all structures electronically and mechanically applied procedures and strategies to distinguish, break down and collect or receive physical content as PC printed records, transcribed content, tag and even magazines. Despite the fact that a considerable measure of headways in OCR innovation have changed the world yet at the same time it can't be used with the Human Reading Ability. Fig. 1. Arrangement of OCR framework Optical character Reader application can be essentially of two sorts on the web and disconnected. Disconnected OCR application takes the info picture first and after that procedure it and perceive. Online OCR applications actualize character acknowledgment in a hurry which implies that the framework receives when you give the framework input. Disconnected mode requests all content in one go though the online mode which prepare the content one by one as the information is given. The created framework executes online mode in acknowledgment module. OCR is a wide point with measurable field of utilizations. This paper incorporates collecting PC printed archives, written by hand message, number plate of an auto and Sudoku Image. The replication of Human Reading Ability into the machine was a fantasy however with the approach of OCR innovation this fantasy has transformed into a reality. The issue definition for this research paper is to make a framework that performs and portrays the working of Optical Character Recognition System and some of its applications. Through this paper we have displayed an answer for the issues of physical archive accumulating, retyping of records and to contain the essential usage and working of the OCR algorithm. This framework can be utilized by business, enterprises, understudies and government offices for different purposes. The framework expanded enormously which benefits the offices and businesses for computerization.

A small OCR sequence and advances in this field is portrayed as beneath:-

1) Beginning and starting of the OCR framework: During the year 1950 information prepared was got to be altered and assumed a vital part in business, firms and ventures. Information was given strength into the PC framework by punch cards. At this point OCR framework were progressively utilized for business reason. The

machine was still primitive, vast, moderate, wasteful and required high portion and the expanding cost

2) First era of OCR framework: The original speaks to the period from 1960 to 1965 where the framework used were obliged with constrained character shapes
3) Second era of OCR framework: The second era speaks to the framework somewhere around 1960 and mid-1970. This period of machine could collect machine and hand printed characters however the character set was extremely constrained
4) Third era of OCR framework: Systems in centre 1975s speak to the third era OCR framework. These could beat the issues of past period machine yet at the same time was extremely costly and constrained to business utilize as it were
5) OCR today: Now the OCR is utilized monetarily furthermore accessible for the general population. Despite the fact that not utilized as a part of our every day schedule.

OCR framework have been monetarily practical yet at the same time the OCR framework are not ready to coordinate the human examining capacity and it is still being developed mode.

1) Image Acquisition: This is the way toward obtaining computerized design pictures from physical record or picture that can be further subjected to picture preparing procedures and can be further utilized as a part of the application. The picture database utilized for testing this framework are procured through a scanner and an advanced camera. The picture database utilized for Car Number Plate Reader involves 50 auto pictures, Sudoku Image Solving database comprises of 55 pictures in its dataset. The pictures for tag discovery are considered at three distinct introductions i.e. front side of the auto, back side of the auto, somewhat tilted auto picture as information
2) Pre-preparing: The obtained picture is of no utilization on the off chance that it is not subjected to the pre-handling stage. This stage make the picture appropriate for further examination and work to be performed on the picture. Pre-preparing incorporates a few application particular methods. A portion of the systems utilized in this venture are:-

Binarization - The procedure of binarization includes transformation of examined picture into a grayscale picture which is changed over into paired or bi-level picture. This progression is essential to recognize the foundation which is set to white and closer view, the content to be received which is set to dark.

Loud and bad noise Removal - Noise in picture concludes to the unpredictable movement or variety in luminance or shade of the picture. Event of loud noise in picture is typically unavoidable because of scanner, printer or computerized camera mistake. In this way it might happen that the loud noise gets changed over into dark closer view spot amid binarization and which could be mixed up for some info character. Along these lines it is vital to examine or check through the loud noise. For this distinctive sorts of channels relying on the application are connected. In this progression we have utilized Median Filter to expel salt and pepper loud or noise from the picture if there should arise an occurrence of Car number plate acknowledgment and manually written character picture acknowledgment.

Edge Detection: The resultant picture from the past retrieval is subjected to edge identification for recognizing the edges and limits of the character in the picture. We have connected Operator in this progression. At that point the shade of the picture is modified so that the limits are set apart in dark and hence include extraction should be possible effortlessly.

Line, Word and Character Segmentation: The given information content picture should be fragmented into lines then to words and individual characters, so that these individual characters can be received effectively. This step includes Connected Component Labelling which marks the parts on the premise of pixel availability. This stage additionally uses flat and vertical projection strategies to fragment the content into lines and words. For this we plot a flat and a vertical histogram of the picture having crests (White) and valleys (Black). Valleys portrays the limits of line and words. Hand composing acknowledgment and Computer printed report Recognition Modules requires each of the three division. Auto Number Plate Reader and Sudoku Solver. In this paper we have composed Online Handwriting Recognition, Car License Plate Identification and Recognition, Image Sudoku Solver and Image to Text Changer as a major aspect of OCR applications. The exactness or the effectiveness of an OCR framework is straightforwardly extent to the nature of the picture given as info. Non-Distinguishable foundation, high commotion picture may debase the productivity of an OCR framework. The outcomes and investigation of the methods portrayed are tasteful and the exactness of the framework is up to the stamp. The created framework is quick, solid and exact. The proposed strategy can supplant the current approach to some degree and can be placed being used in different zones of uses. The modules planned are easy to use and simple to utilize. Different sorts of pictures are tried to get the sought yield. This work is fruitful in meeting every one of its destinations of making an OCR framework and describe or portray some of its applications. As innovation advances and develops every day, a most recent revelation or creation may get to be distinctly outdated sooner than one can anticipate. In this way in future with more propel methods and calculations, this work can be further enhanced by including new elements.

Identifying and solving Sudoku using computer vision

In [9] author proposed the simulated vision unquestionably which gives suggestion to picture preparing, these pictures are just the crude material of a very much extensive science, the same as endeavours to imitate human perceptual capacities, Sudoku is a standout amongst the most prominent puzzle recreations ever, therefore was fascinating to apply the programming information to illuminate a typical every day challenge, the objective of Sudoku is to fill a 9 × 9 network with numbers so that every line, segment and 3 × 3 segment contain the greater part of the digits somewhere around 1 and 9. The point of the work is demonstrate that with an information of programming with a webcam in a pc is conceivable to apply systems of picture handling to distinguish the Sudoku range and explain it. All the work was produced in C ++ utilizing Open CV, QT creator Ide, Tesseract Libraries and the code to explain the Sudoku is open source ought to be considered as further improvement figure. A few papers look at the impact of methodologies when solving or improving Sudoku grids or puzzles. People have been considering machines with the capacity to have or execute and translate printed literary reports, so they can be consequently changed over into a substitute medium or organization [8]. Optical Character Recognition (OCR) let machines to perceive characters. For instance, OCR is utilized to perceive optically prepared printed character in tag number which depends on format coordinating.

A research of the OCR history and the different methods utilized for OCR advancement as a part of the sequential request is being done in. Enlarged reality permits the client to see this present reality, increasing it with superimposed virtual question, it is a procedure for including and supplementing advanced substance over genuine utilizing PCs and cell phones. A typical practice is to utilize the cameras on these gadgets to catch video, draw extra substance over it and show the outcome on the screen of a similar gadget; a down to earth case is to interpret signs and marks starting with one dialect then onto the next. Increased reality innovation can be utilized as a part of a few sorts of utilizations concern the fields of excitement, upkeep, assembling and medicinal care. These days the PCs has a great deal of handling ability even is regular that has an incorporated webcam, the need of clear and justifiable pictures conveyed humanity to make methods to enhance a picture, if important diminish loud noise, disturbances, change the brilliance, shading transformation, and so on. Our work builds up a program to recognize the Sudoku range explain it and print the arrangement with enlarged reality on a screen. The calculation comprise in distinguish the Region of Interest (ROI) for yield. We exchange just the areas of numbers in the framework and distinguish if has a number or is unfilled, if a number is available no activity is done yet in the event that the network position is void, consequently we put a zero. A short time later we read every line one by one to grasp or retrieve what numbers are available to explain the Sudoku. In the following stage we change the picture in numbers so as to the open the source code which explains the confusion. At last with the arrangement got, be continue to print on the show the outcome, accordingly acquiring a program which settles with enlarged reality this mainstream diversion. To peruse the numbers we utilize Tesseract in light of the fact that is assumed or precomputed the most precise open source Optical character acknowledgment (OCR) motor accessible. It was one of the main 3 motors in the 1995 University of Nevada Las Vegas Accuracy test. Somewhere around 1995 and 2006 it had little work done on it, however from that point forward it has been enhanced broadly by Google. It is discharged under the Apache License 2.0. And in addition Open CV, Tesseract is supported or strengthened by the most house stages, the library deals with Linux, Windows and Mac OSX, it can likewise incorporated for other plat- frames, including Android and the IPhone. We utilized the code as a part of where illuminate Sudoku by one by one causing or ruling out numbers to exhaust cells. Before containing a number, the calculation checks whether it is sheltered to allocate. It fundamentally watches that a similar number is not present in current line, current segment and current 3 × 3 sub matrix. Subsequent to checking for wellbeing, it allots the number, and recursively checks whether this task prompts to an answer or not. On the off chance that the task doesn't prompt to an answer, then the calculation attempt next number for current purge cell. Also, if none of number (1 to 9) prompt to arrangement, we return false or negative value here.

Approaching with the test of settling a side interest, particularly Sudoku, we take after a few rules; which are engaged to effectively decide the zone of enthusiasm, in this specific case was sheet acknowledgment; once it is accurately recognized we perform morphological operations to keep just the imperative data (numbers), next most important thing is to translate the data read and create the outcome to compose it on screen progressively.

Video Acquisition:

The video must be procure to build up the application, there are for the most part two types of preparing:

1) Getting outlines straightforwardly from a camcorder or
2) From a video document beforehand recorded. For this application we utilize the main alternative. Video Capture is a component of OpenCV, this capacity permits read the video, either from a record show on the PC or from a camera that is to state it peruses a picture from a cushion in memory.

Pre-preparing:

Change Colour Space Open CV utilizes as a matter of course the BGR coding (Blue, Green, Red), so we continued to change the picture to space shading HSV (Hue, Saturation, Value), where you can isolate the channels staying away from issues with shine since this shading space has an alternate channel to the splendour, this can't be in RGB in light of the fact that is appropriated in every layer of shading. The HSV shading model characterizes a shading space as far as three constituent parts: Hue is the shading sort, (for example, red, blue, or yellow), Saturation is the intensity of the shading, additionally here and there called the immaculateness.

Morphological Operations:

The means for coming to get a ROI are: utilize Canny capacity that rapidly discover the forms display in the picture, which is an essential strategy to calculate or extract or retrieve valuable basic data from various objects and decrease the measure of information to be handled. The general method for edge detection incorporates: discovery of edges with low error rate, which implies that the location ought to precisely get whatever number edges appeared in the pictures as could be allowed; the second criteria is that the edge point identified from the administrator ought to precisely limit on the focal point of the edge and the third is that a given edge in the picture ought to just be stamped once, and where conceivable or connectable, picture loud noise or corruption ought not make false edges. Next tough task or important thing is enlarging and dissolving that permit complement or expel little particles. A goal is lessen the camera's loud noise and disturbance and get a decent complexity of visual attributes displays on the picture Dilation and disintegration are two important morphological operations. Expansion adds pixels to the boundaries of articles in a picture, while vice versa expels pixels on question limits. The number of pixels included or excluded from the items in a picture relies on upon the state and size of the organizing component used to handle the picture.

Find ROI

In the past procedure we get a picture with all around characterized edges so now we simply search for forms and in this manner separate little zones, concentrating all handling in the biggest calculated region in the picture. Once guaranteed the Sudoku range, we can concentrate in the particular data in light of the fact that every one of the edges caught by the webcam contain a considerable measure of pointless data that moderates the calculation. Select the form having the biggest zone. With green rectangle turned a red rectangle without pivoted is produced, which serves to hold when the calculation sends the request to take a photo of the video.

Drawing a Grid

We draw the network for two reasons: distinguish the right area of the Sudoku grid or puzzle to figure out if has or not number, if has number the area, it is put away and in the event that it is unfilled a next task permits us to put a zero for the arrangement Sudoku numerical calculation and the second reason is on account of permit give an enlarged reality. The sides of the rectangle are given by the calculation, we isolate every side into 9 sections, we create x and y places of every purpose of division and after the division is connected its comparing position as an afterthought with lines. This division outwardly observed as ventures. Now we concluded what our advantage range is and we have recognized each of boxes.

Categorising level of problems in Sudoku solving technique

In [10] creators have utilized two strategies for tackling and demonstrating Sudoku issues, in particular, Constraint Satisfaction Problem (CSP) and Satisfiability Problem (SAT) approaches. To this impact we characterize the Generalized Sudoku Problem (GSP), where districts can be of rectangular shape, issues can be of any request, and arrangement presence is not ensured. Concerning the most pessimistic scenario many-sided quality, we demonstrate that GSP with piece locales of m lines and n segments with n=m is NP-Complete. For concentrate the experimental hardness of GSP, we characterize a progression of occurrence generators that contrast in the adjusting level they ensure between the essential or urgent things of the issue, by finely controlling how the gaps are appropriated in the cells of the GSP. Tentatively, we demonstrate that the more adjusted are the urgent things, the higher the total number of sides and nature of explaining the GSP occurrences, and that GSP is harder than the Quasigroup Finish Problem (QFP), an issue summed up by GSP. At long last, we give an investigation of the connection between's spine factors—factors with a similar esteem in every one of the arrangements of a case—and hardness of GSP. They propose a speculation of the Sudoku issue that thus, is a QFP with extra imperatives, however with specific square district requirements that include or absorb QFP. The proposed speculation augments Sudoku in a few bearings. To start with, rather than normal Sudoku, a Generalized Sudoku Problem (GSP) may receive any subjective size. Second, piece areas don't should be square shaped.

GSP additionally has a decent scaling conduct by permitting rectangular piece districts. At the point when utilizing just square piece locales, after a 5 × 5 Sudoku (25 sections crosswise over and 25 pushes high), the following issue accessible is a 6 × 6 Sudoku, that is, 36 pushes by 36 segments. The hop in hardness between a 5 × 5 and a 6 × 6 is immense and difficult, hence leaving scientists with either simple issues or difficult issues. Rectangular formed issues hardness, as can be found in our experimental outcomes, fits in the middle of those square melded sizes. For benchmarking purposes they provide a modest bunch of size and alternatives to browse, offering more conceivable or retrievable outcomes on hardness. A Latin Square (LS), or Quasigroup, of request s, is an s × s grid, with each of its 2 (two) cells loaded with one of the images, to such an extent that no image is created in succession or segment. A legitimate finish Generalized Sudoku (GS) of requests on s images, is a LS of request s with the extra confinement that every image happens precisely once in every square district. A square locale is a touching arrangement of s pre-characterized cells; piece areas don't cover, and there are precisely s square districts in a GS of request s. On account of square piece locales, every square area is a √s × √s network (s must be a square number); on account of rectangular piece districts, every piece area is an m × n grid (m lines and n sections) with m× n = s. At that point, a GS with m× n piece locales will have n area lines and m district sections (as an outlining case, demonstrates a GS structure with m = 2 and n = 3). Once a substantial GS is produced, to make a GSWHP we should punch openings to be filled. The least difficult technique to expel values from the GS is to pick which cells will be expelled arbitrarily. Really, this was the strategy utilized as a part of Lewis (2007) to make Sudoku examples. This makes issues that is destined to be, typically, simpler to explain than if we pick these gaps taking after an example. This is genuine particularly when such an example is adjusted, that will be, that the quantity of openings in each line, section or square district is the same (or fundamentally the same as). We will introduce here three techniques to punch gaps, every one logically giving a more refined example, and we will see later, in the trial comes about, this refined adjustment intensely impacts issue hardness.

Singly balanced:

In the first place, we consider the adjusted example utilized as a part of Kautz et al. (2001) for QWHP occasions that we call here separately adjusted. In an independently adjusted example we have, when receivable, a similar number of openings in each line and segment of the Sudoku. Given the aggregate number of openings h, we can disperse q = h/s gaps in every line and segment of the Sudoku utilizing a calculation for customary bipartite diagram era, in view of a Markov Chain calculation (Kannan et al. 1997). Watch that a gap design with q openings in each line and each section is proportional to a q-standard bipartite chart (R ∪ C, E), with R the arrangement of lines of the Sudoku and C its arrangement of segments and (r,c) ∈ E demonstrates that there is a gap in position (r, c) of the Sudoku. We move along the Markov chain, where each state is an example that fulfils that the quantity of gaps in every line and segment is the same, utilizing a "switch" (Kannan et al. 1997). A switch is characterized by two passages (i, j),(i ,j) of the GS case, with the end goal that there is an opening in (i, j) and (i ,j) yet not in (i, j) and (i, j). On the off chance that we change both openings to positions (i, j) and (i, j) the line and segment gap number does not change. At the point when q = h/s is not a number, we can at present create, with a similar calculation, a practically adjusted example with a bipartite diagram where the level of the vertices is either q or q + 1. For this situation, we make the underlying entire example putting q gaps in each line and segment of the GS. At that point we disperse the rest of the (h mod s) openings by a manner that lacks in selecting (h mod s) extra cells without any lines or sections in like manner.

Doubly adjusted:

Because the dispersion of openings between various pieces can have any kind of effect in the trouble of the issue, we propose another technique that guarantees that the quantity of gaps in each line, segment, and square area will be the same. Our new doubly adjusted technique depends on the Markov chain of, however now every state is a gap example that additionally fulfils that the quantity of gaps is the same in all squares. Along these lines, we utilize this Markov chain, however we improve the quality of the moves to those moves that additionally keep up the quantity of openings present in every square, i.e., moves with the end goal that the pieces of (i, j) and (i ,j) are either in a similar piece push or a similar square section, or both. Such moves dependably exist, notwithstanding for an entire example with just a single opening in every square. With this, we have the code neat and clean and optimized and faster to read, in Algorithm 1 for producing an opening example H with q = h/s gaps in every line, section and square, utilizing a GS S (i, j) (with images {1,...,s}) to make the underlying example considering every image as a gap, and after that performing t travels through the Markov bind so as to test from the arrangement of getting doubly adjusted gap designs.

Fully balanced:

Our last balanced method, which we call it as fully balanced, is a generalization of the previous one. Here, we also push the number of holes in each row and column inside each block to be the same, whenever possible. So, this method produces a fully balanced hole pattern if the block regions are square (n = m), if the total number of holes h satisfies that q1 = h/s is an integer (the number of holes in each region, row and column of the Sudoku) and if q2 = q1/n is also an integer (the number of holes in each row and column inside any block region). For that reason, in this model we restrict regions to be square (so n = √s), but we do not restrict the number of holes, so in general we will not always get a fully balanced pattern. If these conditions are met, as what we surely need in every block is a whole pattern that is singly balanced inside the block, the following simple code generates a fully balanced Sudoku.

With this, we have the code neat and clean and optimized and faster to read, in Algorithm 1 for producing an opening example H with q = h/s gaps in every line, section and square, utilizing a GS S (i, j) (with images {1,...,s}) to make the underlying example considering every image as a hole, and after that performing t traversals through the Markov attach so as to test from the arrangement of getting doubly adjusted hole designs.

CHAPTER – 3

SYSTEM DEVELOPMENT

Our calculation is outlined particularly to illuminate a Sudoku grid or puzzle containing a 9 x 9 grid containing numbers and blank spaces. The principle assignment of the calculation is separated into two areas:

Firstly, it ought to precisely find the network position in the picture while dealing with issues of foundation puzzle, scaling, interpretation, turn and point of view skew.

The second part of the calculation should then find those positions in the matrix where there are numbers and remember them with accuracy. A recursive backtracking calculation will then settle the puzzle.

The solved solution with output written in the blank spaces is sent back to the mobile phone using Wi-Fi.

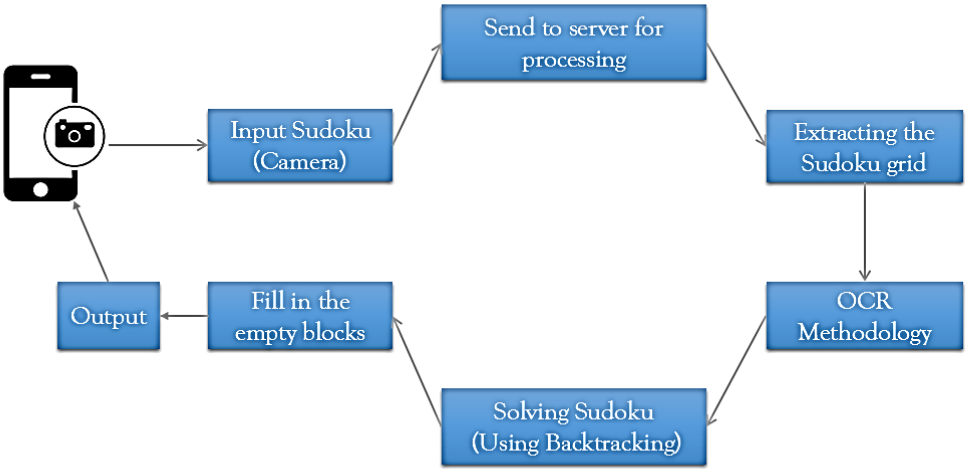

Send to server for processing
Input Sudoku (Camera)
Extracting the Sudoku grid
OCR Methodology
Solving Sudoku (Using Backtracking)
Fill in the empty blocks
Output

Figure 3.1: System Architecture

3.1 Extracting the puzzle grid

When we have taken a picture, as appeared in figure-3.1, containing the Sudoku lattice we first change over it to a dim scale picture, as appeared in figure-3.2, which is then the contribution to other pre-handling strategies so that the network can be precisely found. The means that the image we took are talked about underneath:

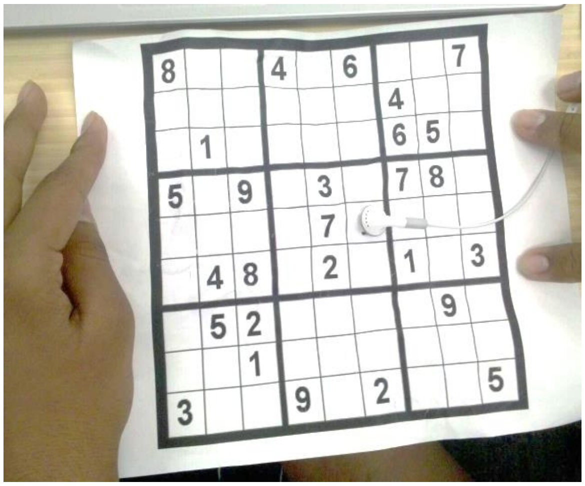


Figure 3.2: Original image of Sudoku Puzzle

3.1.1 Smoothing the image and Adaptive thresholding

The information picture is initially examined and checked or analysed with a low-pass Gaussian channel of 5x5 window measure. The purpose behind this is it smoothens out sharp projections and other high recurrence segments in the picture. Likewise it helps in showing signs of improvement results in this way.

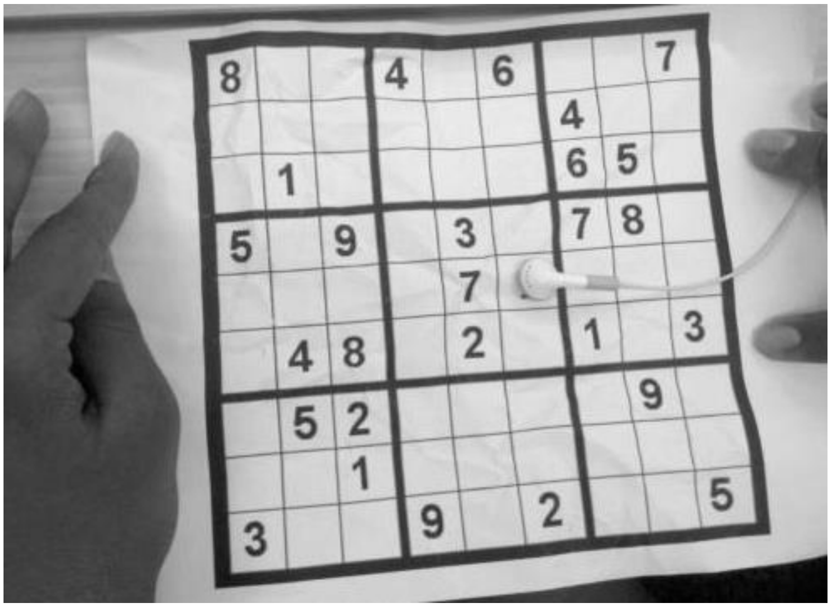


Figure 3.3: Gray Scale Image of Sudoku

Thresholding, in this way bringing about hearty lattice corner extraction. It was additionally watched that performing versatile thresholding on the picture without the smoothing channel gave broken digits. In the wake of unclear arrangements, the digits got were spotless and finish, which could be utilized for digit acknowledgment. Once smoothed, the picture is gone through an Adaptive thresholding calculation. The thinking behind this is there might be a brightening slope (for instance, because of shadows of the human catching the pictures with the cell phone), over the picture and worldwide thresholding will totally wreck the subtle elements of the framework and digits making it difficult to concentrate it. Applying window-wise thresholding helps in confining the impact in this way adequately drawing out the subtle elements which are ideal for our calculation. The window measure utilized by us for Adaptive thresholding was heuristically been 30 percent of the inexact size of each 1x1 square in the obtained 9 x 9 matrix of the grid. This is on the grounds that applying thresholding in this way over the whole network helps in getting better subtle elements in the numbers subsequently making our calculation for perceiving the digits more strong.

Figure-3.3 demonstrates the outcome got in the wake of smoothing and versatile binarization.

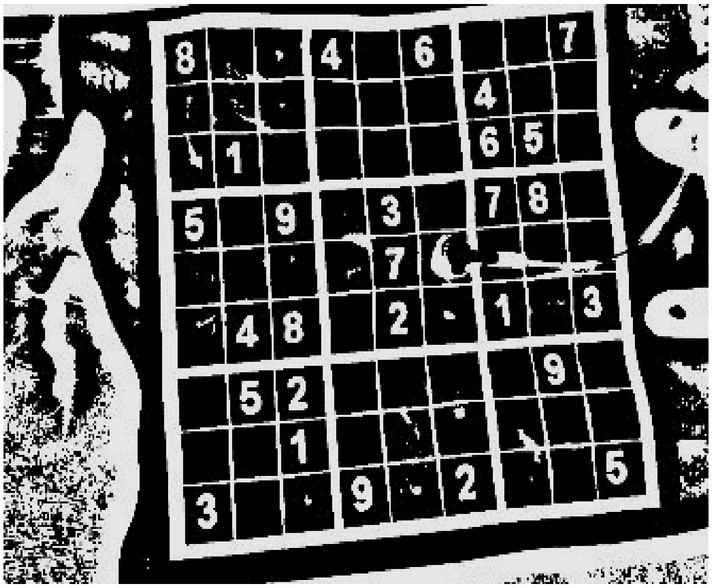


Figure 3.4: After applying Gaussian Blur and Adaptive Thresholding

3.1.2 Isolating the grid from background clutter of the image

A vital and important part of the calculation is to accurately recognize the area of the Sudoku Grid in the given picture. To accomplish this, we depend on the idea of Connected Components. The matrix without anyone else's input is an associated part and our theory is that it is the biggest associated segment in the picture subjected to specific limitations. These imperatives were presented in the wake of recognizing a few weaknesses in our past approach. The theory that the Sudoku framework is the biggest associated part is right as a rule yet bombed in a few. These cases were the place the picture was portrayed with the nearness of blobs outside the Sudoku framework that had high pixel thickness. In such cases, in spite of the fact that the lattice is one of the biggest associated segments, it won't not be the biggest one. To defeat this issue, while performing important filling calculation, a procedure to distinguish every associated part utilizing 4 network as a part of the picture, we monitor the biggest associated segment, i.e. the one with the greatest pixel thickness and very thick. This Connected part needs to satisfy certain different limitations. These urgent things are in direct relationship to our necessity that the Sudoku Grid has a structure and width of no less than 30 individually. This limitation gave great outcomes for all intents and purposes and precluded undesirable jumble which had the capability of being mis-identified as the framework segment. To make our limitations nearer to recognizing the genuine matrix, we force another confinement. The sides of the associated segment found ought to have its corner focuses introduce in the picture, notwithstanding that reality that they ought not cover with the edges of the picture.

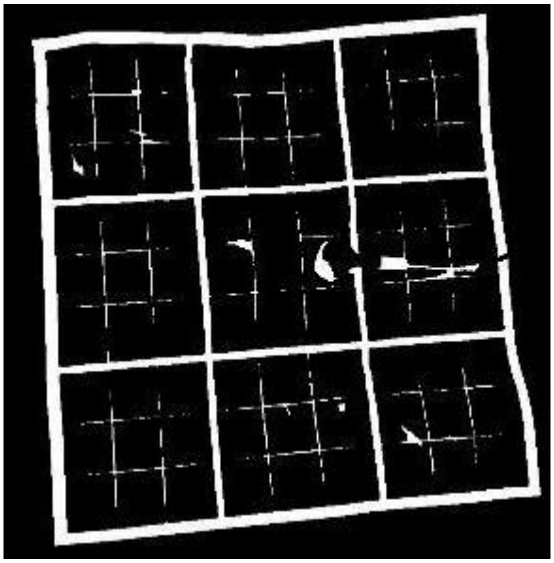


Figure 3.5: Extracted Grid

This approach, gave very good results in terms of grid detection. It removes any clutter outside the grid and isolates the Sudoku grid accurately and eases the rest of our algorithm. The extracted grid is shown in Figure-3.4.

3.2 Recognition of Digits

Now that we obtained the extracted grid, our next task is to locate the digit positions in the puzzle grid and recognize them. For this we use tesseract OCR engine.

3.2.1 Tesseract OCR engine

Tesseract is an open source optical character acknowledgment (OCR) motor initially created at Hewlett-Packard somewhere around 1985 and 1995 for recognizing digits or numbers in an image. Since HP had autonomously created page design investigation innovation that was utilized as a part of items, (and in this way not discharged for open-source) Tesseract never required its own page format examination. Tesseract in this way accept its information is a twofold picture.

Preparing takes after a customary well-ordered pipeline. The initial step is an associated part examination in which layouts of the segments are put away. This was a computationally costly plan choice at the time, yet had a critical favourable position: by examination of the settling of blueprints, and the quantity of youngster and grandchild traces, it is easy to distinguish opposite content and remember it as effortlessly as dark on-white content. Tesseract was most likely the main OCR motor ready to handle white-on-dark content so unimportantly. At this stage, diagrams are assembled, simply by settling, into Blobs. Blobs are sorted out into content lines, and the lines and districts are broke down for settled pitch or corresponding content. Content lines are broken into words contrastingly as per the sort of character dispersing. And then words are used for character recognition, so this process is called optical character recognition.

Acknowledgment then continues as a two-pass handle. In the primary pass, an endeavour is made to gain every word thus we can obtain each character from a word. Every word that is attractive is passed to a versatile classifier as preparing information. The versatile classifier then gets an opportunity to all the more precisely obtained content let down the page. Since the versatile classifier may have learned something valuable past the point where it is possible to make a commitment close to the highest point of the page, a moment ignore is run in the page, in which words that were not obtained all around correctly are perceived once more.

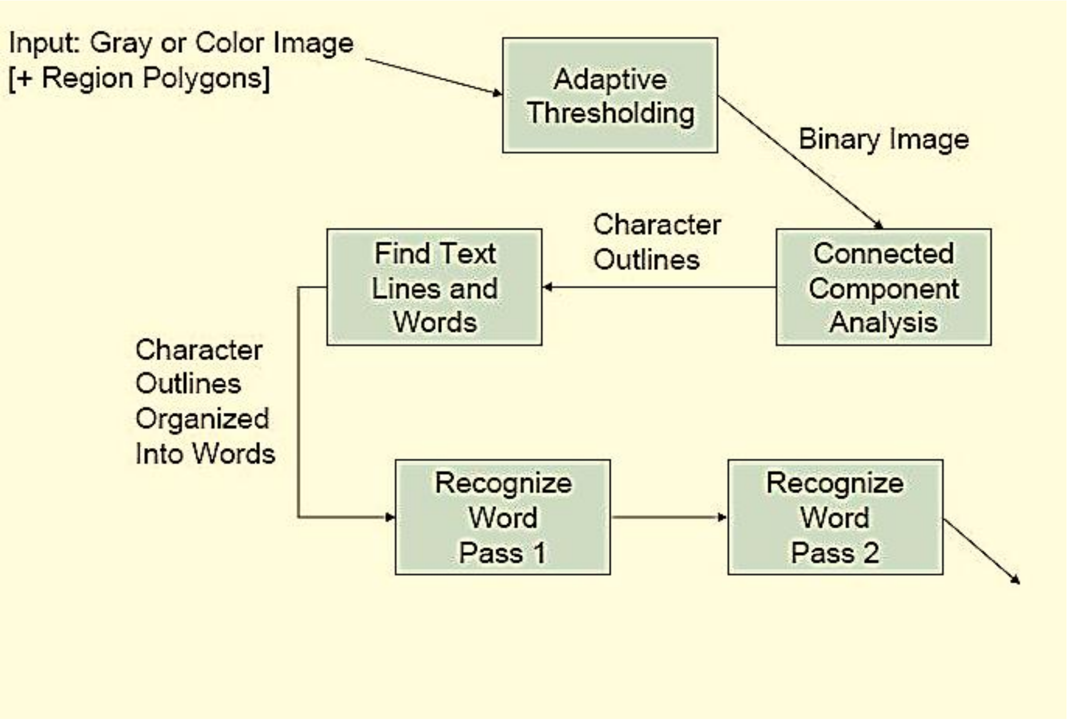

Input: Gray or Color Image
[+ Region Polygons]
Adaptive Thresholding
Binary Image
Connected Component Analysis
Character Outlines
Find Text Lines and Words
Character Outlines Organized Into Words
Recognize Word Pass 1
Recognize Word Pass 2

Figure 3.6: Architecture of tesseract

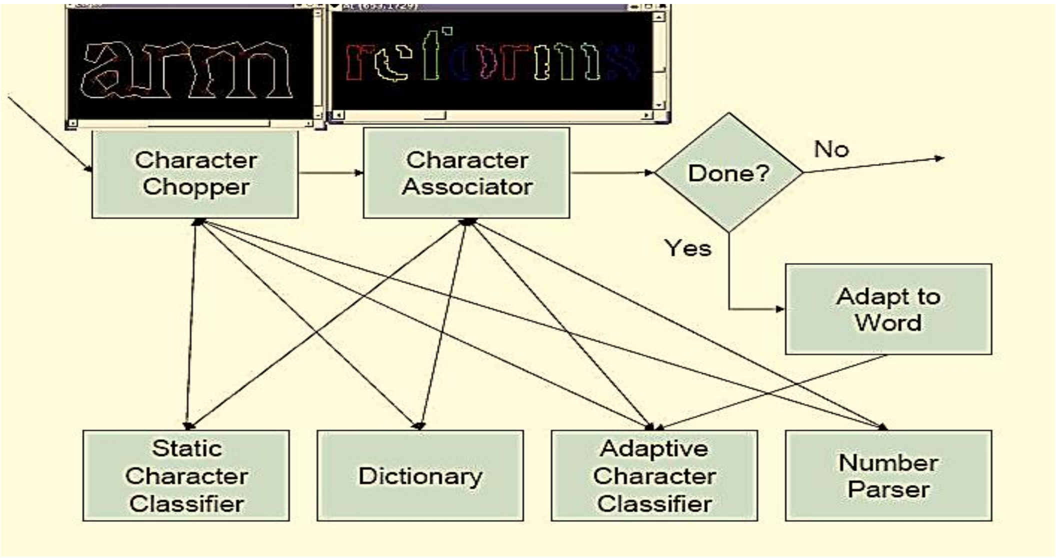


Figure 3.7: Architecture of tesseract

3.2.1.1 Working of tesseract

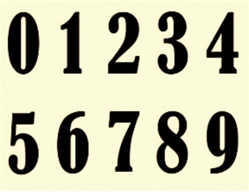
The accompanying is a brief diagram of how Tesseract functions:

1. Outlines are investigated and put away
2. Outlines are assembled as Blobs
3. Blobs are sorted out into content lines
4. Text lines are broken into words and digits are retrieved
5. First go of acknowledgment process endeavours to gain every word
6. Satisfactory words go to optimised mentor
7. Lessons learned by optimised mentor utilized in a moment pass, which guarantees obtaining the words that were not obtained attractively in the main pass
8. Fuzzy spaces settled and message checked for little tops
9. Advanced writings are yielded and returned

Tesseract employment techniques:

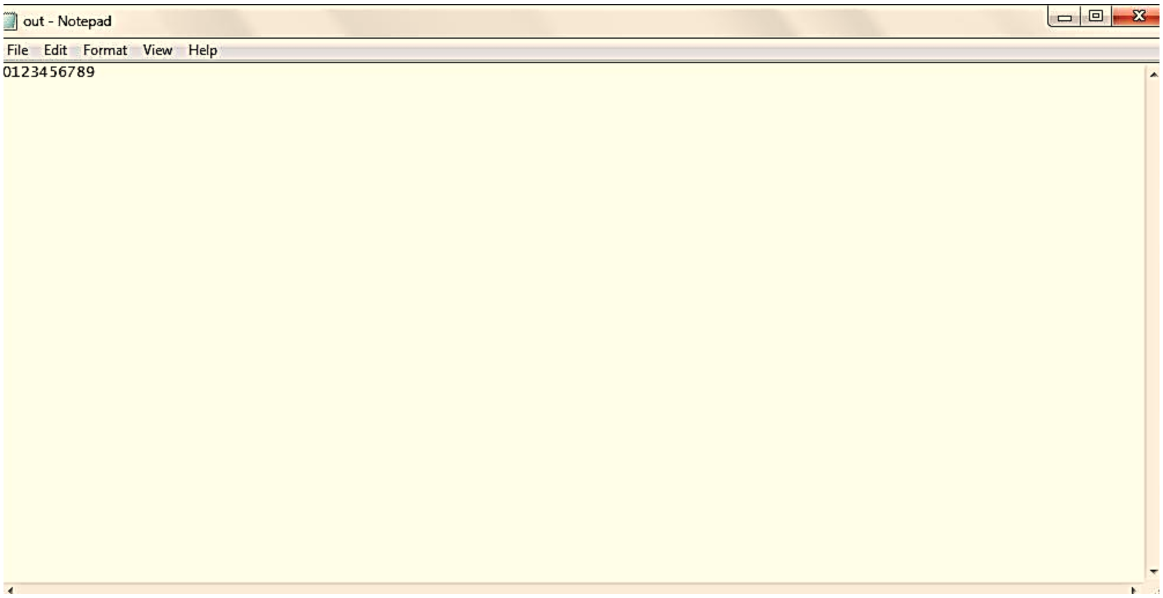
- Algorithms for recognizing content lines from a changed or distorted page
- Algorithms for recognizing relative and non-corresponding words (a corresponding word is a word where every one of the letters are a identical width)
- Algorithms for cleaving joined characters and for partner broken characters
- Language investigation to distinguish the in all probability word framed by a set of characters

Two character classifiers: a static classifier, and a versatile classifier which utilizes preparing information, and which is better at recognizing upper and lower case letters



0 1 2 3 4
5 6 7 8 9

Figure 3.8: Input image



```
0123456789
```

Figure 3.9: Recognised digits output in notepad

3.3 Sudoku Solving

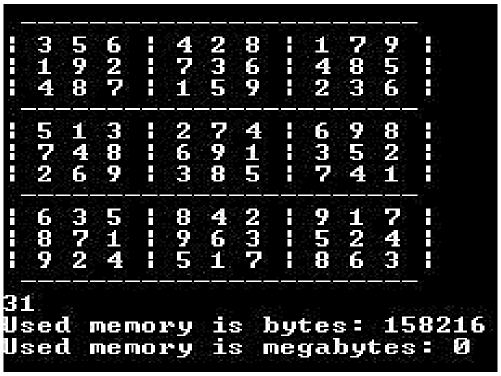
After the digits have been successfully recognized, the Sudoku puzzle can now be solved using a proper algorithm. We have used Recursive backtracking to solve the Sudoku puzzle

3.3.1 Recursive backtracking

The calculation to grasp or make us understand the Sudoku depends on a recursive backtracking procedure. We store the numbers acquired from the network in the picture, in a two dimensional matrix (say 9 x 9 or 16 x 16) with the number 0 allocated to clear or blank cells. To get an answer for the matrix, we recognize a clear area in the lattice. In view of Sudoku standards, we distinguish a substantial task by emphasizing through the numbers 1 to 9. We then attempt to recursively calculate the framework with this new number position. On the off chance that there are no more network areas which should be filled, it means that the framework is calculated, and we give back the changed lattice back to the principle calculation. Then again, on the off chance that we neglect to recursively discover an answer for a framework, we backtrack and attempt an alternate number from 1 to 9 for an area by incrementing the current number and traverse the matrix again for correct solution. In the event that all blends of number assignments are exhausted without finding an answer or a proper solution is not found, we finish up the network is unsolvable and in this manner the capacity will return false. For our motivations anyway, we accept that the picture taken by the client is a substantial Sudoku puzzle, which is solvable and less distortion and clear enough to recognize all the digits by OCR and our calculation will be independent of the difficulty level of the puzzle whether it’s hard or easy, just the complexity will be affected nothing much.

3.3.1.1 Algorithm

```
Find row, col of an unassigned cell, if there is none return true
For digits from 1 to 9
    a) If there is no conflict for digit at row, col
            • assign digit to row, col and recursively try fill in rest of grid
    b) If recursion successful, return true
    c) Else, remove digit and try another continuing from the removed digit
If all digits have been tried and nothing worked, return false
```



```
 -----------------------------
¦ 3 5 6 ¦ 4 2 8 ¦ 1 7 9 ¦
¦ 1 9 2 ¦ 7 3 6 ¦ 4 8 5 ¦
¦ 4 8 7 ¦ 1 5 9 ¦ 2 3 6 ¦
 -----------------------------
¦ 5 1 3 ¦ 2 7 4 ¦ 6 9 8 ¦
¦ 7 4 8 ¦ 6 9 1 ¦ 3 5 2 ¦
¦ 2 6 9 ¦ 3 8 5 ¦ 7 4 1 ¦
 -----------------------------
¦ 6 3 5 ¦ 8 4 2 ¦ 9 1 7 ¦
¦ 8 7 1 ¦ 9 6 3 ¦ 5 2 4 ¦
¦ 9 2 4 ¦ 5 1 7 ¦ 8 6 3 ¦
 -----------------------------
31
Used memory is bytes: 158216
Used memory is megabytes: 0
```

Figure 3.10: Solution using backtracking

3.4 DFD

3.4.1 DFD LEVEL 0

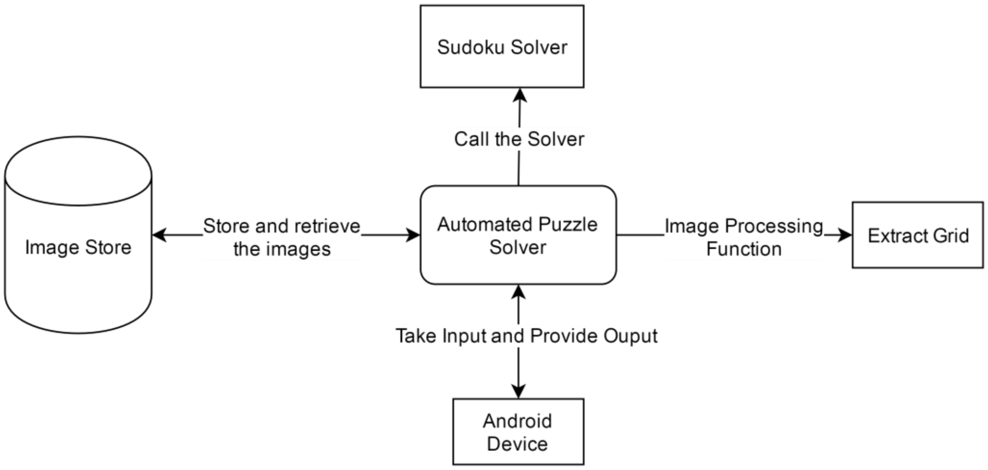


Figure 3.11: DFD Level 0 representing an overview of the whole project

3.4.2 DFD LEVEL 1

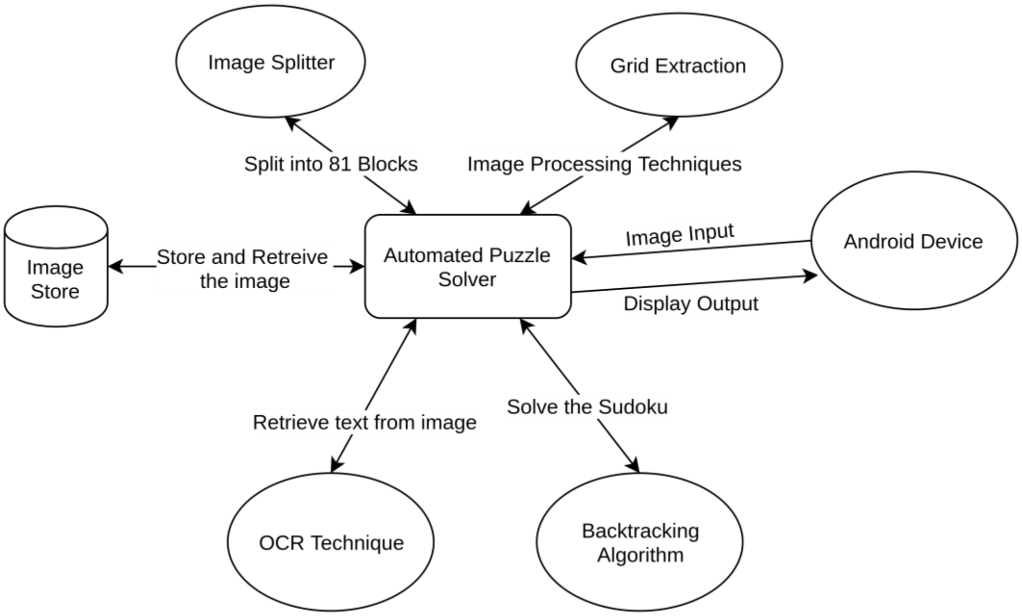


Figure 3.12: DFD Level 1 representing details of each module

3.4.3 DFD LEVEL 2

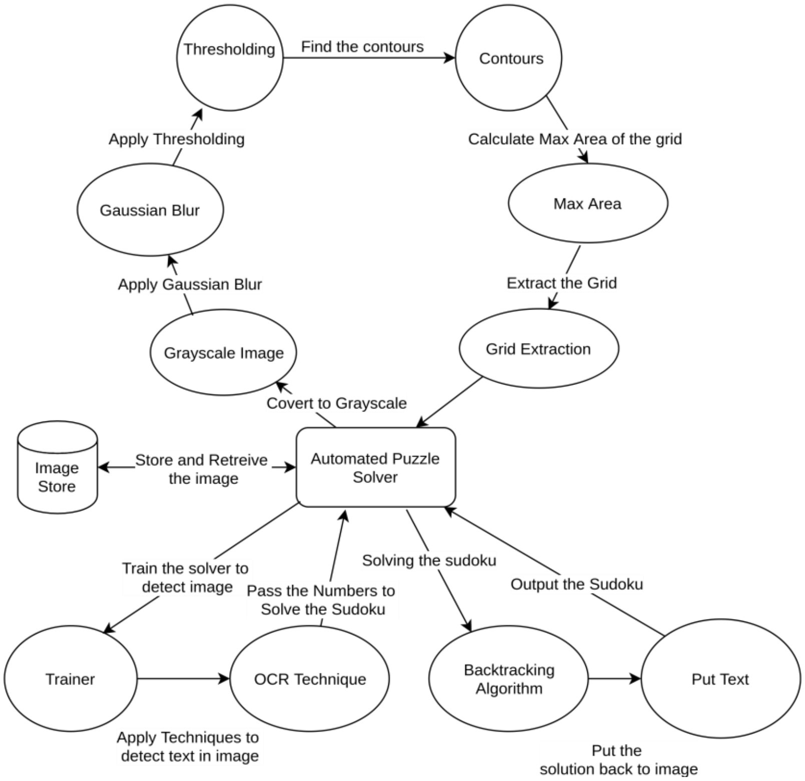


Figure 3.13: DFD Level 2 representing connections between different modules

CHAPTER – 4

PERFORMANCE ANALYSIS

4.1 Other algorithms for Sudoku Solving

4.1.1 Dancing Links:

Moving Links, otherwise called DLX, is the strategy recommended by Donald Knuth to effectively execute his Algorithm X. Calculation X is a recursive, nondeterministic, great depth to start with, backtracking calculation that discovers all answers for the correct cover issue. A portion of the better-known correct cover issues incorporate Sudoku puzzle. The name Dancing Links originates from the way the calculation works, as emphases of the calculation cause the connections to "move" with accomplice interfaces in order to look like a move. Knuth credits Khei Noshita and Hiroshi Hitotsumatsu with creating the thought in 1979, however it is his paper which has advanced it. To make the less quality turn into more quality more quantity, the network of Sudoku is expected to change over the issue into an Exact Cover Problem, we have to conclude what the lines and sections speak to.

The segments speak to the urgent thing of the surprise. Every number accompanies its own arrangement of important things that is the numbers here in our case for Sudoku puzzle. In this manner there are size to the power 2 * 4 sections where SIZE is the quantity of competitors/lines/cols there are in the Sudoku Puzzle. In a 4x4, this would be 64 segments. In a 9 x 9, this would be 324 sections.

The lines speak to each and every retrievable position for each number. Consequently, there are SIZE to the power 3 lines. In a 4x4, this would be 64 sections. In a 9x9, this would be 729 columns. Every column would speak to just a single applicant position. Thusly, just 4 1s will be in the column, speaking to the requirements of that position.

The inadequate framework for a 4x4 Sudoku puzzle is seen in: 4 x 4 and 64 x 64 puzzle. The 9 x 9 matrix or lattice is unrealistic to make by hand. Given beginning positions in the network, those columns will be incorporated into the answer and secured. At that point the Search calculation will deliver the answers for the grid.

4.1.2 Brute Force:

A few specialists have created PC programs that will traverse Sudoku grids utilizing a savage constrain calculation. In spite of the fact that it has been built up that around 6.67 x 1021 last frameworks exist, utilizing a beast drive PC calculation can be a useful strategy to settle confusions if the code is all around planned.

A beast constrain calculation visits the void cells in some request, filling in digits successively from the accessible decisions, or backtracking (evacuating fizzled decisions) when a deadlock is come to. For instance, a constraint program would illuminate a puzzle by putting the digit "1" in the main cell and checking in the event that it is permitted to be there. On the off chance that there are no significance (checking line, section, and box requirements) then the calculation advances to the following cell, and spots a "1" in that phone. At the point when checking for correct results, it is found that the "1" is not permitted, so the esteem is progressed to a "2". On the off chance that a cell is found where none of the 9 digits is permitted, then the calculation leaves that cell clear and moves back to the past cell. The esteem in that cell is then expanded to "2". The calculation is traversed again for correct results of Sudoku cell until a correct number or answer for every one of the 81 cells is found. A case of this is:

1. randomly ruling or eliminating out numbers to the clear cells in the framework
2. calculate the quantity of blunders or mistakes and call it probability
3. "shuffle" these embedded numbers around the network until the quantity of oversights is reduced to 0

An answer for the puzzle will then have been found. Approaches for rearranging the numbers incorporate strengthening, hereditary calculation and table inquiry. Random probability distribution - based streamlining calculations are known to be very quick, however they are maybe not as quick as some dynamic based procedures. Not at all like the last mentioned be that as it may, enhancement calculations don't really make issues to be rationale resolvable, giving them the possibility to take care of a more extensive scope of issue occasion. It is likewise correct to express a Sudoku as a whole number straight programming issue. Such methodologies appear to draw near to an answer rapidly, and can then utilize spreading towards the end. This is how a Sudoku can be solved using brute force by calculating probability distribution of a number which can come at that particular place or cell. Hence this method is Brute force as it calculates all possible permutations and probability of getting there. The Simplex calculation appears to be ready to handle circumstances without any arrangements or different arrangements great.

4.2 Comparison of different algorithms

The following graph shows the comparison between backtracking and dancing links in terms of memory used for different difficulty levels of Sudoku puzzle.

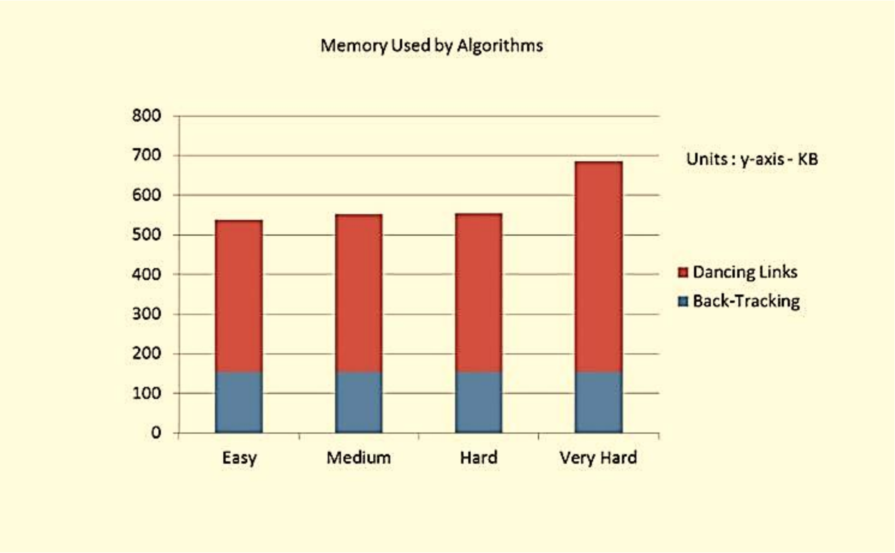


Figure 4.1: Memory taken by different algorithms

4.2.1 Time taken by algorithms

The following graphs shows the amount of time taken by the algorithm for different difficulty levels of Sudoku puzzle.

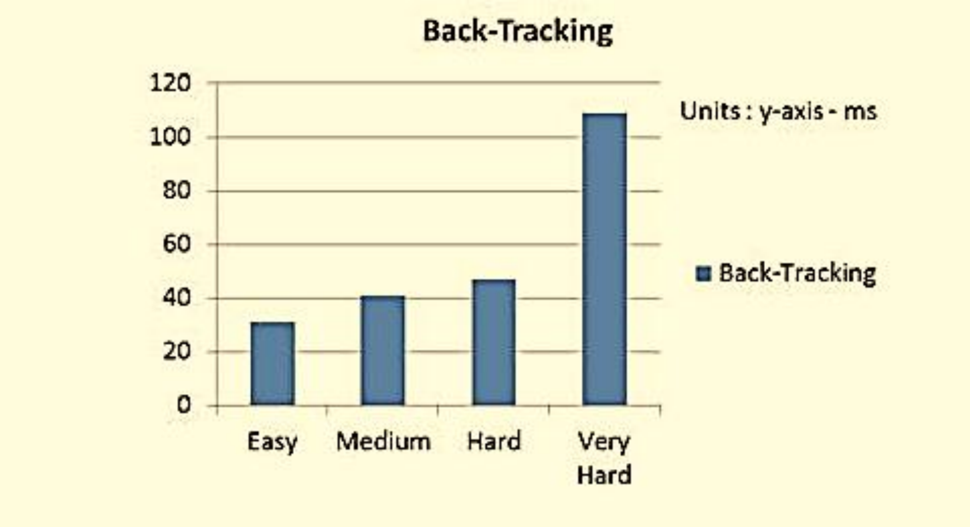


Figure 4.2: Time taken by Backtracking Algorithm

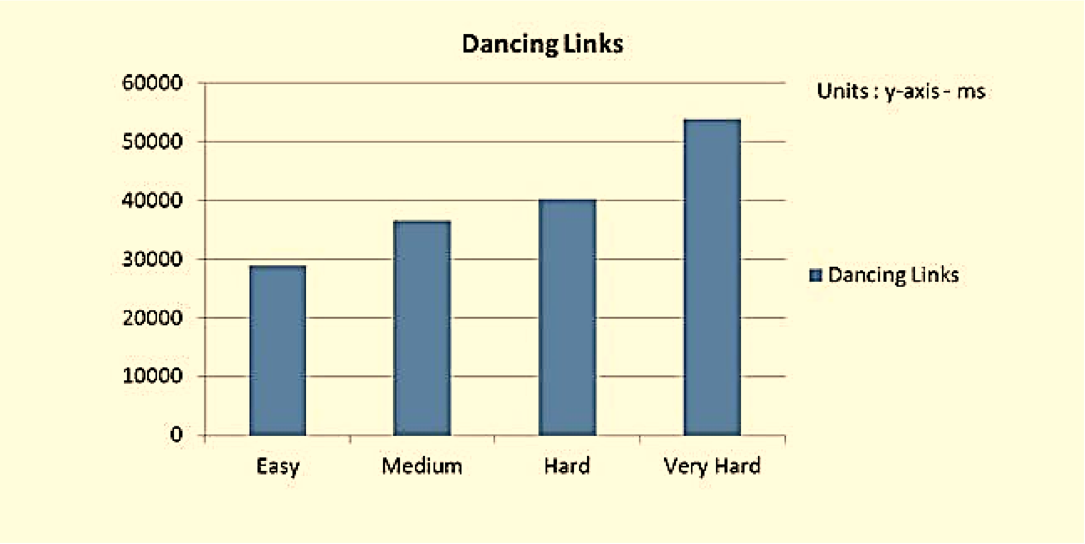
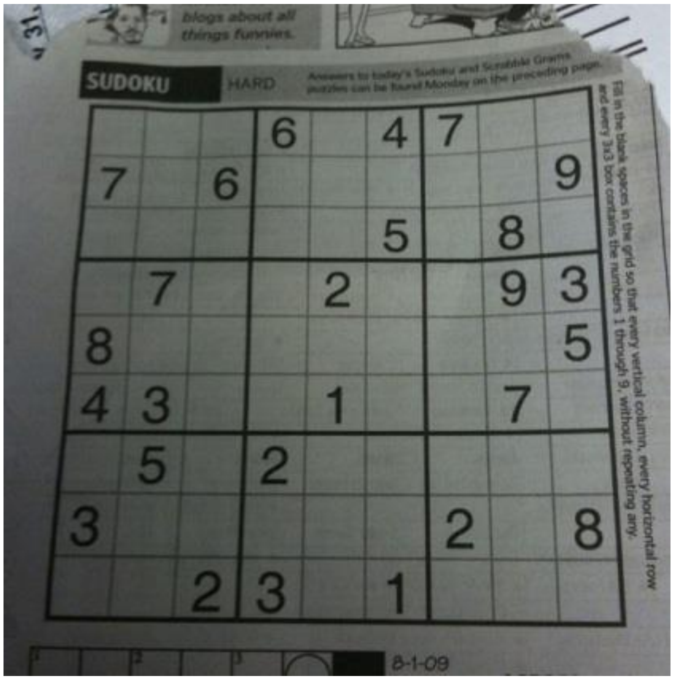


Figure 4.3: Time taken by Dancing Links

4.3 Output at various stages

1. Sudoku puzzle



SUDOKU HARD

Answers to today's Sudoku and Scrabble Grams puzzles can be found Monday on the preceding page.

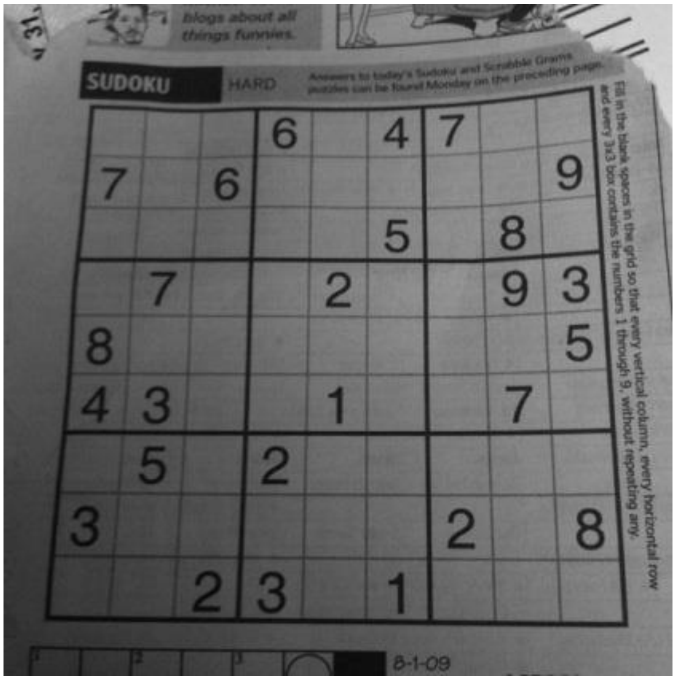
Fill in the blank spaces in the grid so that every vertical column, every horizontal row and every 3x3 box contains the numbers 1 through 9, without repeating any.

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

8-1-09

Figure 4.4: Sudoku Puzzle

2. Converting to gray scale

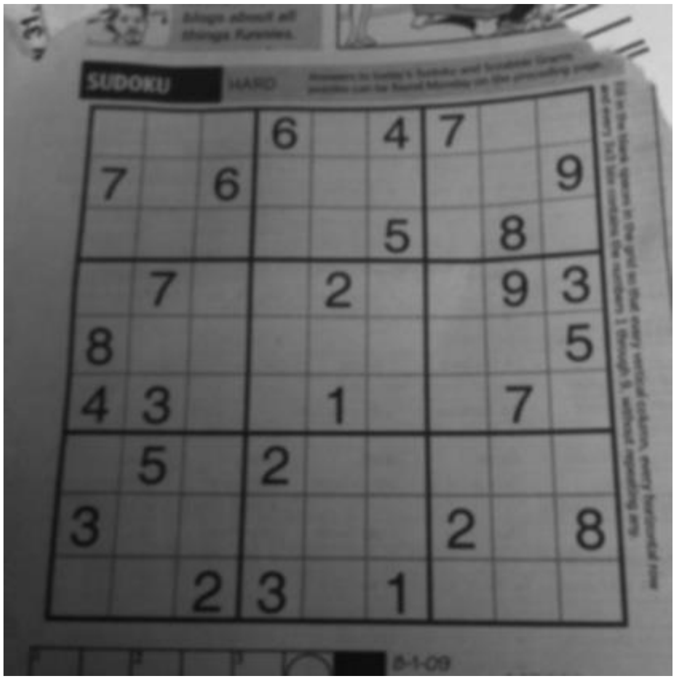


31
blogs about all things funnies.
SUDOKU
HARD
Answers to today's Sudoku and Scrabble Grams puzzles can be found Monday on the preceding page.
Fill in the blank spaces in the grid so that every vertical column, every horizontal row and every 3x3 box contains the numbers 1 through 9, without repeating any.
8-1-09

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

Figure 4.5: Gray Scale

3. Applying Gaussian blur



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

Figure 4.6: Gaussian Blur

4. Applying adaptive thresholding

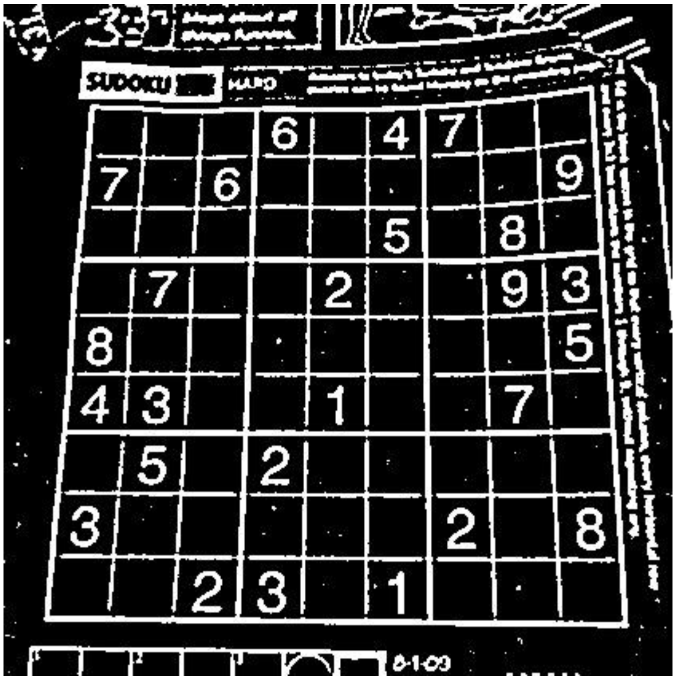


Figure 4.7: Adaptive Thresholding

5. Extracting the grid

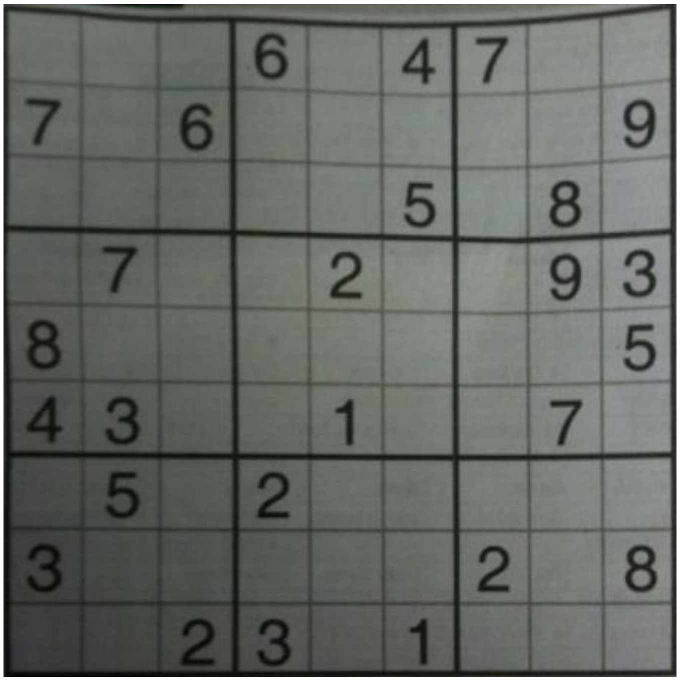
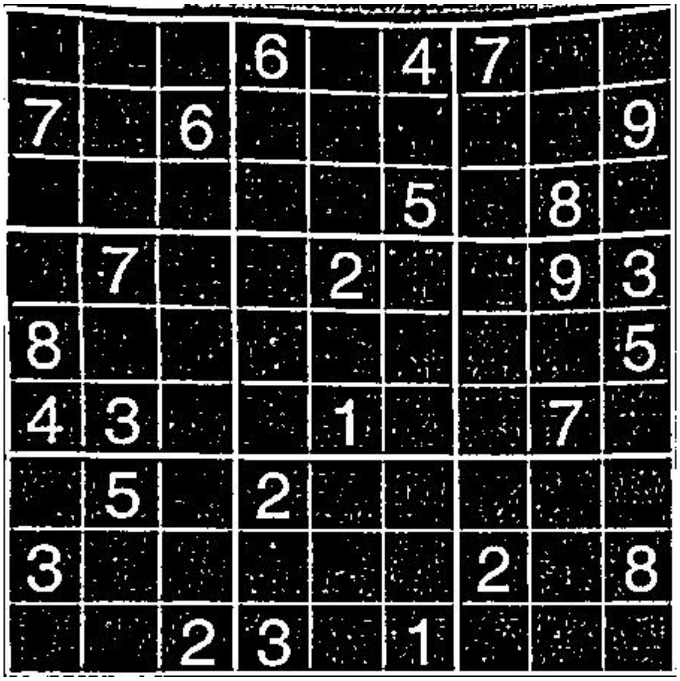


Figure 4.8: Extracting Grid

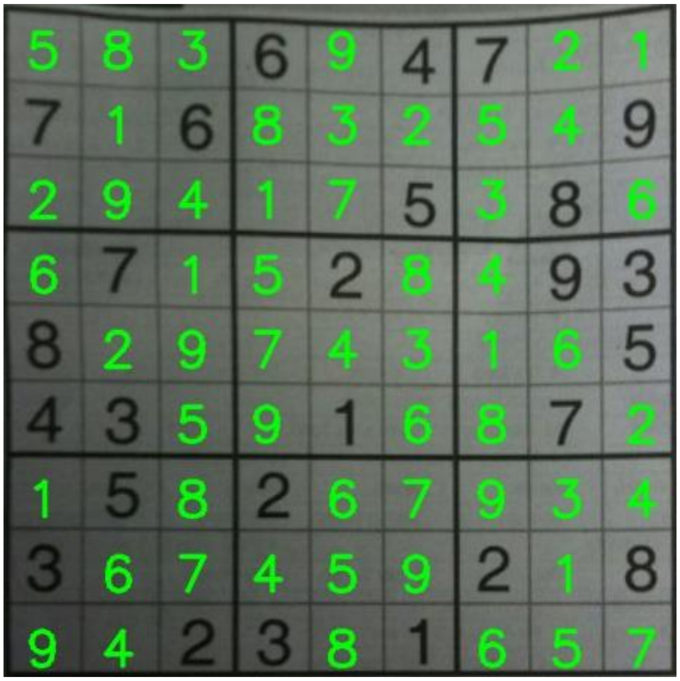
6. Thresholding the extracted grid



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

Figure 4.9: Thresholding the grid again

7. Extracting Numbers and writing output



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

Figure 4.10: Extracting numbers and writing solution

CHAPTER – 5

CONCLUSION

5.1 Conclusions

We display a Smart Sudoku Solver that can easily solve unsolved Sudoku pictures with little measure of viewpoint taken from any camera. Likewise illuminated changes over the pictures are dealt with. The calculation can likewise give arrangement in instances of serious pivot, for example, when the Sudoku Puzzle is totally distorted. Since the size of the picture likewise shifts from picture to picture, our calculation effectively handle these issues. Likewise the tesseract OCR has been able to do effective retrieval of all the digits. Since the time and memory prerequisite contrasts for different calculations, we tried these calculations for different difficulty levels, extending from simple level to hard level. In addition the diverse calculations were analysed for their proficiency and the most optimised one was utilized to illuminate the Sudoku Puzzle.

5.2 Applications and Contributions

5.2.1 A Two-Layer Steganography Scheme Using Sudoku for Digital Images

Steganography is a technique to extract the secret data in the digital images without getting any unexpected notices to hackers or attackers. LSB replacement directly replaces the secret bits with LSB bit plane. Unfortunately, it is not secure as it cannot resist the visual attacks and statistic detection. In 2009, Lin et al. showed a novel data embedding scheme by using the idea of Sudoku. Lin et al.'s scheme not only improved the performances of traditional LSB-based steganography schemes, but also resist against steganalysis. In this paper, a steganography scheme based on Lin et al.'s scheme is explored. Experimental results show that the proposed scheme can improve the visualising quality of stego-images. In addition, the average of hiding capacity is 1.29 bpp (bits per pixel). Thus, it confirms that our proposed scheme provides higher abstraction capacity than that of LSB replacement 0.29 bpp. Furthermore, our method shows the positive results to prevent against the visual attacks.

References:

[1] JOHN CANNY, “A Computational Approach to Edge Detection”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986.

[2] https://web.stanford.edu/class/ee368/Project_Spring_1415/Reports/Wang.pdf

[3] ARNAB K. MAJI, SUDIPTA ROY, RAJAT K. PAL , “A Novel Algorithmic approach for solving Sudoku puzzle in Guessed Free Manner”, EUROPEAN ACADEMIC RESEARCH, VOL. I, ISSUE 6/ SEPEMBER 2013

[4] Manav B. Sanghavi, Aniket K. Rupani, Mahek S. Maniar, Sai Deepthi Pabba, “Solving Sudoku from an Image using Modular Architecture Approach”, International Journal on Recent and Innovation Trends in Computing and Communication , Volume: 3 Issue: 3, March 2015

[5] Ricardo Soto, Broderick Crawford, Cristian Galleguillos , Kathleen Crawford, Fernando Paredes , “A Filtering Technique for Helping to Solve Sudoku Problems” , Springer, 21 July 2015.

[6] Akash Dutta , Arunabha Ghosh , “Development of a Character Recognition Software to solve a Sudoku Puzzle”, IEEE 2016

[7] J. Sauvola, M. PietikaKinen , “Adaptive document image binarization” , Elsevier 1999

[8] Anubhav Jain, Jaya Sharma , “Classification and Interpretation of Characters in Multi-Application OCR System” , IEEE 2014

[9] Darío José Mendoza Chipantasi , Nancy del Rocío Velasco Erazo , “Augmented Reality for Automatic Identification and Solving Sudoku Puzzles Based on Computer Vision” , Springer, 04 August 2015.

[10]Carlos Ansótegu,Ramón Béjar,Cèsar Fernández,Carla Gomes,Carles Mateu , “Generating highly balanced sudoku problems as hard problems” , Springer 2011