

SUDOKU SOLVER with OPENCV

A Project Work

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

Submitted By

Shivam

19021180046

Nityansh Bhardwaj

19021011394

Under the supervision of
Mr. Subhash Chandra Gupta



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA – 201306**



**SCHOOL OF COMPUTING AND SCIENCE AND
ENGINEERING
BONAFIDE CERTIFICATE**

Certified that this project report **“SUDOKU SOLVER with OPENCV”** is the bonafide work of **“Shivam(19SCSE1180051) , Nityansh bhardwaj (19SCSE1180094)”** who carried out the project work under my supervision.

**Mr. Subhash Chandra Gupta
Assistant Professor
School of Computing
Science & Engineering**

ABSTRACT

OpenCV is a library of Python bindings that is generally used to solve problems related to computer vision. It is a cross-stage library accessible in a wide assortment of programming dialects, like C++, Python, Java, and so forth.

OpenCV can be utilized to handle pictures and recordings to distinguish articles, faces, or even the penmanship of a human. We use vector space and perform numerical procedure on these highlights to distinguish picture example and its different highlights. The capabilities of the app include being able to detect and recognize such a puzzle by getting feed from web cam. It should be able to solve the game in real time and fill the empty squares with the correct digits, displaying them over the feed from the camera on the screen in the corresponding positions.

ACKNOWLEDGEMENT

Working on this project on "SUDOKU SOLVER with OPENCV" was a source of immense knowledge to us. We would like to express my sincere gratitude to "Mr. A. Daniel" for his guidance and valuable support thought out the course of this project work. We acknowledge with a deep sense of gratitude, the encouragement and inspiration received from our faculty members and colleagues. We would also like to thank our parents for their love and support.

Piyush Kumar Gupta, Abhinav Krishna Singh, Shivam

LIST OF FIGURES

Fig.1.	Overview
Fig.2-5	Tools
Fig.6	Preprocess the image
Fig.7	Adding threshold
Fig.8	Find the big contour
Fig.9	Print detected digits
Fig.10	Solved Digits
Fig.11	Overlay Solution

TABLE OF CONTENTS

- i. Certificate
- ii. Abstract
- iii. Acknowledgement
- iv. List of Figures

- 1. Introduction
 - OVERALL DESCRIPTION.....1-2
- 2. Principal Components.....3-6
- 3. Proposed Plan.....7
- 4. Problem Formulation.....8-15
- 5. Feasibility Analysis.....16
- 6. Implementation.....17
- 7. Conclusion.....18-19
- 8. References.....20

INTRODUCTION

OVERALL DESCRIPTION-

Sudoku is a logic-based combinatorial puzzle. The game comprises of a 9x9 matrix partitioned into 9 square sub-frameworks (of size 3x3) which we will allude to as squares in this paper. Some of the squares in the grid contain a number from 1 to 9. The player is presented with a partially filled grid and their aim is to fill the rest of the squares with numbers from 1 to 9. The rules are fairly simple: each row, column and block must contain each of these numbers exactly once. As of now, Sudoku puzzles are getting progressively well known among individuals everywhere on the world. The game has gotten famous now in countless nations and numerous designers have attempted to create significantly more confounded and additional intriguing riddles. Today, the game shows up in pretty much every paper, in books and in numerous sites. The best power is an overall calculation than can be applied to any conceivable issue. This calculation creates any potential arrangements until the right answer is found. The accompanying subsections depict the issue proclamation, the motivation behind this task, and the contractions and the definitions.



Fig . 1

The part of PCs in puzzle world is at present ending up being progressively huge, since PCs changed conundrum producers, at this point furthermore puzzle solvers . Allmost all questions are just for diversion. A kind of puzzle is Sudoku (pencil and paper puzzles). The word Sudoku means 'the digits should remain single. Impediment satisfaction might be characterized as a cycle with an assortment of elements having necessities constrained on them that should be satisfied. There are distinctive veritable models where CSP is used, as computerized masterminding and booking of classes for the semester. Addressing Sudoku as a Constraint Satisfaction Problem and use of spread techniques license the puzzles to be understood with polynomial time

thinking. Figure 1 Some works are done to acknowledgment the quantities of Sudoku and addressed it like , in a minigrid based novel procedure is created to disentangle the Sudoku puzzle in speculated free way. By and large, in addressing Sudoku puzzles than requested ones, and will be considered as additional advancement factor. Papers like inspect the impact of stochastic methodologies when tackling Sudoku games. People are considering machines with the ability to "read" and "unravel" printed scholarly records, in demand that they'll be therefore changed over into a substitute medium or design. Optical Character Recognition(OCR) let machines to recognize characters. As a model, OCR is utilized to recognize optically took care of printed character in vehicle place number which is predicated on design organizing . A review of the OCR history and subsequently the different strategies used for OCR improvement inside the sequential solicitation is being depleted . Nowadays the PCs incorporates a great deal of learning capacity even is standard that has a planned webcam, the need of clear and sensible pictures carried mankind to make procedures to upgrade an image, if significant lessening upheaval, dark, alter the magnificence, concealing change, etc Our stir develops a program to recognize the Sudoku district tackle it and print the appropriate response with extended reality on a screen. We move simply the

spaces of numbers inside the lattice and recognize if incorporates a number or is unfilled, if assortment is available no activity is finished except for if the grid position is empty, therefore we put a zero.

The capabilities of the app include being able to detect and recognize such a puzzle by getting feed from web cam. It ought to have the option to address the game continuously and fill the vacant squares with the right digits, showing them over the feed from the camera on the screen in the comparing positions.

Principal Components

The major components for the development of the algorithm are:

- 1) Python
- 2) OpenCV
- 3) Tensorflow
- 4) OCR(optical recognition)
- 5) Sudoku Solver Algorithm

Python Language : We program overall Python language that might be executed on the premier normal working frameworks, as an illustration in Linux, Mac OSX and Windows, the specific work was made in Windows climate.\

Python is a deciphered undeniable level universally useful programming language. Python's plain reasoning underscores code clarity with its remarkable utilization of huge space.

Python is progressively composed and trash gathered. It upholds various programming ideal models, including organized (especially, procedural), object-situated and useful programming.

Python is regularly portrayed as a "batteries included" language because of its far reaching standard library.

Guido van Rossum started dealing with Python in the last part of the 1980s, as a replacement to the ABC programming language, and first delivered it in 1991 as Python 0.9.0.[32] Python 2.0 was delivered in 2000 and presented new highlights, for example, list cognizances and a trash assortment framework utilizing reference checking. Python 3.0 was delivered in 2008 and was a significant correction of the language that isn't totally in reverse viable and much Python 2 code doesn't run unmodified on Python 3. Python 2 was ceased with adaptation 2.7.18 in 2020.

Python reliably positions as perhaps the most mainstream programming dialects.

Some Python Libraries

- TensorFlow
- Scikit-Learn
- Numpy
- Keras
- PyTorch
- LightGBM
- Eli5

- SciPy
- Theano
- Pandas



OpenCV Libraries : OpenCV could be a library of programming limits principally designed for continuous PC vision, made by Intel Russian investigation, was planned for computational adequacy, the library can capitalize on the gear speeding up the essential heterogeneous cycle stage.

OpenCV can be utilized to handle pictures and recordings to recognize articles, faces, or even the penmanship of a human. We use vector space and perform numerical procedure on these highlights to distinguish picture example and its different features



Applications of OpenCV

Some of the applications of OpenCV are:

- Face recognition
- Automated inspection and surveillance
- Vehicle counting on highways along with their speeds
- Anomaly detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control object

recognition.

Optical Character Recognition (OCR):

Optical character recognition or **optical character reader (OCR)** is the electronic or mechanical change of pictures of composed, manually written or printed text into machine-encoded text, regardless of whether from an examined report, a photograph of an archive, a scene-photograph (for instance the content on signs and announcements in a scene photograph) or from caption text superimposed on a picture (for instance: from a transmission).

Broadly utilized as a type of information section from printed paper information records – regardless of whether visa reports, solicitations, bank explanations, mechanized receipts, business cards, mail, printouts of static-information, or any appropriate documentation – it is a typical strategy for digitizing printed messages so they can be electronically altered, looked, put away more minimally, showed on-line, and utilized in machine cycles like psychological figuring, machine interpretation, (removed) text-to-discourse, key information and text mining. OCR is a field of examination in design acknowledgment, computerized reasoning and PC vision.

In order to scrutinize the numbers we use OpenCV on the grounds that it perhaps the first exact open source Optical person affirmation (OCR) engine available. Gotten together with theLeptonica Image Processing Library it's anything but an

enormous style of picture configurations and convert them to message in excess of 60 dialects. It had been one in every one of the greatest 3 motors inside the 1995 University of Nevada city Accuracy test. Some place in the scope of 1995 and 2006 it had little work done on that, notwithstanding from that point forward it's been improved generally by Google. It released under the Apache License 2.0 , likewise as OpenCV is upheld by the foremost commons stages, the library chips away at Linux, Windows and Mac OSX, it may likewise ordered for other plat-structures, including Android and in like manner the iPhone.

TensorFlow:

TensorFlow is a start to finish open source stage for AI. It has a complete, adaptable biological system of instruments, libraries and local area assets that allows specialists to push the cutting edge in ML and designers effectively fabricate and send ML controlled applications.

TensorFlow ecosystem

TensorFlow provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

Robust ML production anywhere

Effectively prepare and convey models in the cloud, on-prem, in the program, or on-gadget regardless language you use.

Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faste.

For JavaScript

Use TensorFlow.js to make new AI models and send existing models with JavaScript.

For Mobile & IoT

Run surmising with TensorFlow Lite on versatile and inserted gadgets like Android, iOS, Edge TPU, and Raspberry Pi.

For Production

Deploy a production-ready ML pipeline for training and inference using TensorFlow Extended (TFX).

Application of tensorflow

1. Image Recognition

It's quite possibly the most mainstream Uses of TensorFlow. It is utilized by Mobile organizations, online media, and other telecom houses. Picture acknowledgment comprises of pixel and example coordinating to distinguish the picture and its parts.

2. Voice Recognition

TensorFlow has critical use in voice acknowledgment

frameworks like Telecom, Mobile organizations, security frameworks, web indexes, and so on It utilizes the voice acknowledgment frameworks for providing orders, performing activities and giving contributions without utilizing consoles, mouse. It is finished utilizing Automatic discourse acknowledgment which is prepared utilizing TensorFlow. These frameworks convert the human voice into text or PC justifiable code by digitizing it.

3. Video Detection

With expanded innovation, organizations and organizations anticipate safer and advanced frameworks. Consequently, the movement discovery is utilized broadly at air terminal security checks, gaming controls, and development location. Here employments of TensorFlow incorporate self-driving vehicle frameworks, robotization, and numerous auto machines.

4. Text-based applications

The instant messages, responses, remarks, tweets, stock outcomes and so on are a method for information. This preparing of information is finished utilizing TensorFlow for the examination reason and arriving at the normal deals. We do it utilizing various methods like supposition examination, a pack of words and some

more. This can assist with discovering the danger related with any association by disentangling the words utilized in messages.



KERAS:

Keras is a profound learning API written in Python, running on top of the AI stage TensorFlow. It was created with an emphasis on empowering quick experimentation. Having the option to go from thought to result however quick as conceivable may be critical to doing great examination.

Keras is:

- **Simple** -- but not simplistic. Keras lessens engineer psychological burden to free you to zero in on the pieces of the issue that truly matter.
- **Flexible** -- Keras adopts the principle of *progressive disclosure of complexity*: simple workflows should be quick and easy, while arbitrarily advanced workflows should be *possible* via a clear path that builds upon what you've already learned.
- **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

Application of Keras

Keras Applications are profound learning models that are made accessible close by pre-prepared loads. These models can be utilized for expectation, include extraction, and calibrating..

Weights are downloaded automatically when instantiating a model. They are stored at `~/.keras/models/`.

Upon launch, the models will be worked by the picture information design set in your Keras setup record at `~/.keras/keras.json`. For example, assuming you have set `image_data_format=channels_last`, any model stacked from this store will get worked by the TensorFlow information design show, "Tallness Width-Depth".

Pre trained model

Prepared model comprises of two sections model Architecture and model Weights. Model loads are huge document so we need to download and separate the element from ImageNet data set. A portion of the well known pre-prepared models are recorded underneath,

- ResNet
- VGG16
- MobileNet
- InceptionResNetV2
- InceptionV3

Sudoku solver

In sudoku solver we explain Sudoku by individually appointing numbers to purge cells. Prior to doling out assortment, the estimation looks at whether it's protected to give. It in a general sense watches that the indistinguishable number is absent in current line, current section and current 3×3 sub grid. Ensuing to checking for security, it allots the sum, and recursively checks whether this task winds up in an answer. If the errand doesn't cause an answer, by then the estimation endeavor next number for current void cell. Furthermore, if none of number (1 to 9) cause arrangement, we return false.

Proposed Plan :-

- Get the sudoku block in the image — Python OPEN CV

- Crop the sudoku image and change perspective — Python OPEN CV

- Extract the digits in some ordered manner
— Python CNN Model

→ solve the sudoku with sudoku solver function

→ overlap the solution with unsolved sudoku

→ Add some grid

→ stack all image and show in screen

Problem Formulation : -

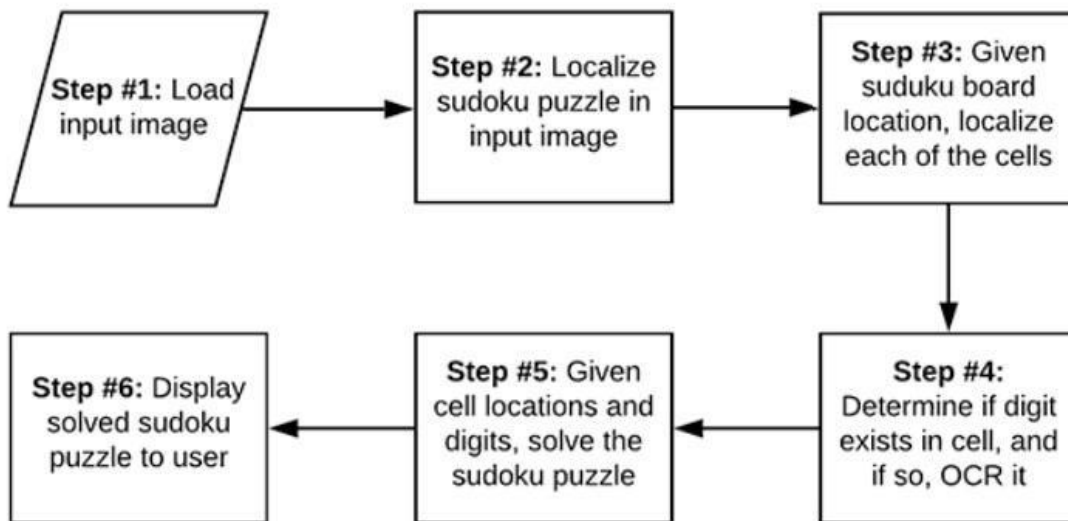
The code for the app is written in Python. In this report, we follow the pipeline of the whole program from getting the feed from the camera to displaying the solution on the screen.

The process consists of:

- Getting a image a prepare it by adding threshold, blur .
- find the contour in the image then draw the contour
- Then find the biggest contour for sudoku puzzle
- Then we split the image and extract the digit from it

- finding the solution with sudokusolver file
- Then overlay the solution into unsolved one
- Stack all image and show it with Cv2.imshow()

To handle the sudoku issue, we at first need to part the issue up into 6 sections,



1. Sudoku Net

Every sudoku puzzle starts with a $N \times N$ grid, where a couple of cells are clear and various cells as of now contain a digit. The objective is to use the data about existing digits to appropriately induce the contrary digits. However, before we tackle sudoku puzzles with openCV , we initially need to execute a neural particular that may deal with OCR'ing the digits on the sudoku puzzle board. We name it SudokuNet

2. Convolutional neural network

CNNs are a development of multi-facet perceptrons, which may learn channels that need to be prepared by the AI models, as tried earlier in [Yann LeCun et al. 1989] using back-engendering. Convolutional networks are essentially applied on visual imagery. Since the planning cycle incorporates getting some answers concerning models from more humble models, A convolutional neural association contains specific concealed layers moreover to the data and yield layers. These specific layers normally fuses convolutional layer with channels which will be learned, redressed long measure layer for use of authorization work, pooling layer for down-testing and setback layer for assurance of discipline for wrong yield. We use Keras with TensorFlow as backend to deal with the CNN model and supply an assessment between the specific CNN utilization outlined by picking particular hyperparameters related with each layer.

3. Training phase

We can start setting up our SudokuNet model, we will use the MNIST instructive record for seeing digits. We start our arrangement with a little unobtrusive pack of imports. Most prominently, we import MNIST enlightening assortment of translated digits, which is joined right with Keras educational assortments module. We generally split our instructive assortment into preparing data and testing data. We add a channel estimation to the digits to show that they are in grayscale. We scale them into an extent of 0 and 1, we use One-hot encoder to seclude and see the numbers with 0 and 1. for example number 3 is addressed as [0,0,0,1,0,0,0,0,0,0] in one-hot encoding. Getting ready is dispatched through a call to the fit technique, when planning is done, we will survey the model and toll our model.

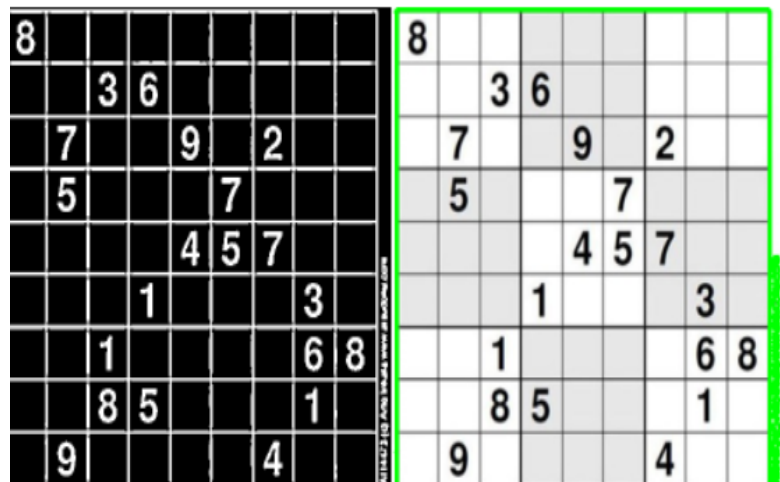
4. Sudoku Puzzle Board

Presently, we have a model that can see digits in the image. Regardless, that digit recognizer doesn't do us much good if we can't discover the sudoku puzzle board in an image. "find_puzzle", finds and thinks the sudoku puzzle board from the data picture. "extract_digits", assesses each cell of the sudoku puzzle board and focuses the digit from the cell. Additionally, we will use "clear_boarder" routine in our "extract_digit" ability to clean up the edges of sudoku cell. Most assignments will be driven with OpenCV with a pinch of help from Numpy and imutils.

5. Find_puzzle

"find_puzzle" work begins things out and recognizes two limits, "input picture" and "investigate". "research" is an optional boolean sign whether to show center advances so you are more probable imagine what's happening in the driving force of our PC vision .

One of our structures will contrast with the outline of the sudoku grid. If the sudoku puzzle isn't found we raise Exception. We will envision what's going on in the motor by drawing the question structure format on the image.



Preprocess the image

```
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
imgBlur = cv2.GaussianBlur(imgGray, (5, 5), 1)
```

```
imgThreshold = cv2.adaptiveThreshold(imgBlur, 255, 1, 1, 11, 2)
```

Adding threshold and blur and contour

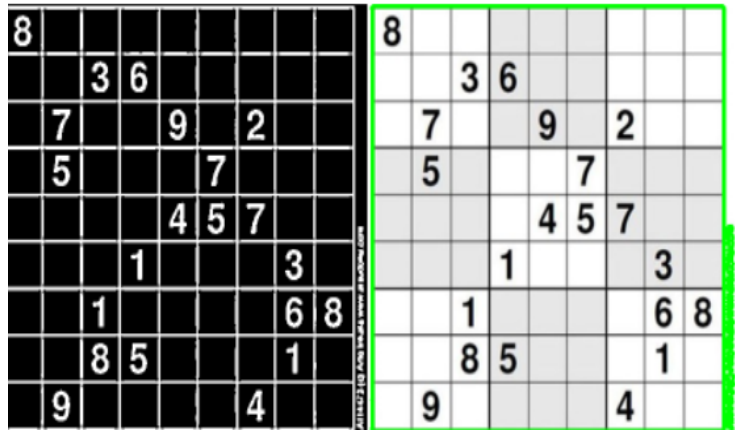


Fig. 7

DRAW ALL DETECTED CONTOURS

With the state of the question nearby, we can deskew the image to secure a hierarchical padded creatures eye point of view on the conundrum. Applying turn perspective change suitably deskews our sudoku puzzle framework, making it much easier for us to choose lines and segments, and cells as we push ahead. This action is performed on the principal RGB and faint pictures. Final image after applying find_puzzle function is shown below

Find the big contour



FIG. 8

```
cv2.drawContours(imgBigContour, biggest, -1, (0, 0, 255), 25)
```

Our `find_puzzle` returns sign of a 2-tuple of the principal RGB picture and grayscale picture after all exercises, including the last four-point perspective change.

6. Extracting Digits

In our past segment, we extricated sudoku puzzle board from a picture with OpenCV. In this portion, we will investigate all of the individual cell in the sudoku board, perceive if there is digit in the cell and expecting thus, eliminate the digit. We describe "extract_image" ability to recognize two limits, "cell" and "investigate"

Cell: A Roi addressing an individual cell of the sudoku puzzle

Investigate: A boolean sign whether midway stem discernments should be showed up on screen.

Our underlying advance, is to clear the front facing region pixels that are reaching the edges of the telephone and find the shapes in edge cell, if no structures are found, we return "None".

From our structures we find the greatest shape by pixle locale and assemble a connected cover .

Disengaging the pixel district in to cover by the area of tyhe cell itself gives us the percentfilled regard.Expecting we don't have commotion in the cell, we apply veil to the thresholded cell. This veil is alternatively appeared on screen.

Print extracted digit



Fig . 9

```
imgDetectedDigits = displayNumbers(imgDetectedDigits,  
numbers, color=(255, 0, 255))
```

7.Sudoku Solver

At this point we are armed with SudokuNet model, warp perspective transform, extracting digits, sudoku solver.

Accounting for every cell in the sudoku puzzle, we loop over rows and columns in a nested fashion.

Inside, we utilize our progression estimates to decide the beginning and finishing directions of the current cell.

Now, we append the coordinates as a tuple to this particular row. Each row will have Nine entries.

Presently, we are prepared to edit out the cell and perceive digits.

Collect the cell from the changed picture and concentrate digit. If the digit isn't none, by then we know there is certifiable digit in the cell, we pre-measure the digit along these lines that we did in preparing and characterize the digit with SudokuNet and update the sudoku puzzle board cluster with the anticipated estimation of the phone.

As should be obvious, first we show sudoku puzzle board as it was deciphered by means of OCR.

To clarify our picture with the arrangement numbers, we just circle over cell areas and the board and unload cell organizes and

figure directions of where text explanation will be drawn.

Presently, we draw each yield digit on our riddle board photograph and show our settled sudoku puzzle picture until any key is squeezed.

Solved Digits



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

Fig. 10

```
imgSolvedDigits=  
displayNumbers(imgSolvedDigits,solvedNumbers)
```

Overlay the Solution



Fig 11

```
imgInvWarpColored = cv2.warpPerspective(imgSolvedDigits,  
matrix, (widthImg, heightImg))
```

```
inv_perspective = cv2.addWeighted(imgInvWarpColored, 1,  
img, 0.5, 1)
```

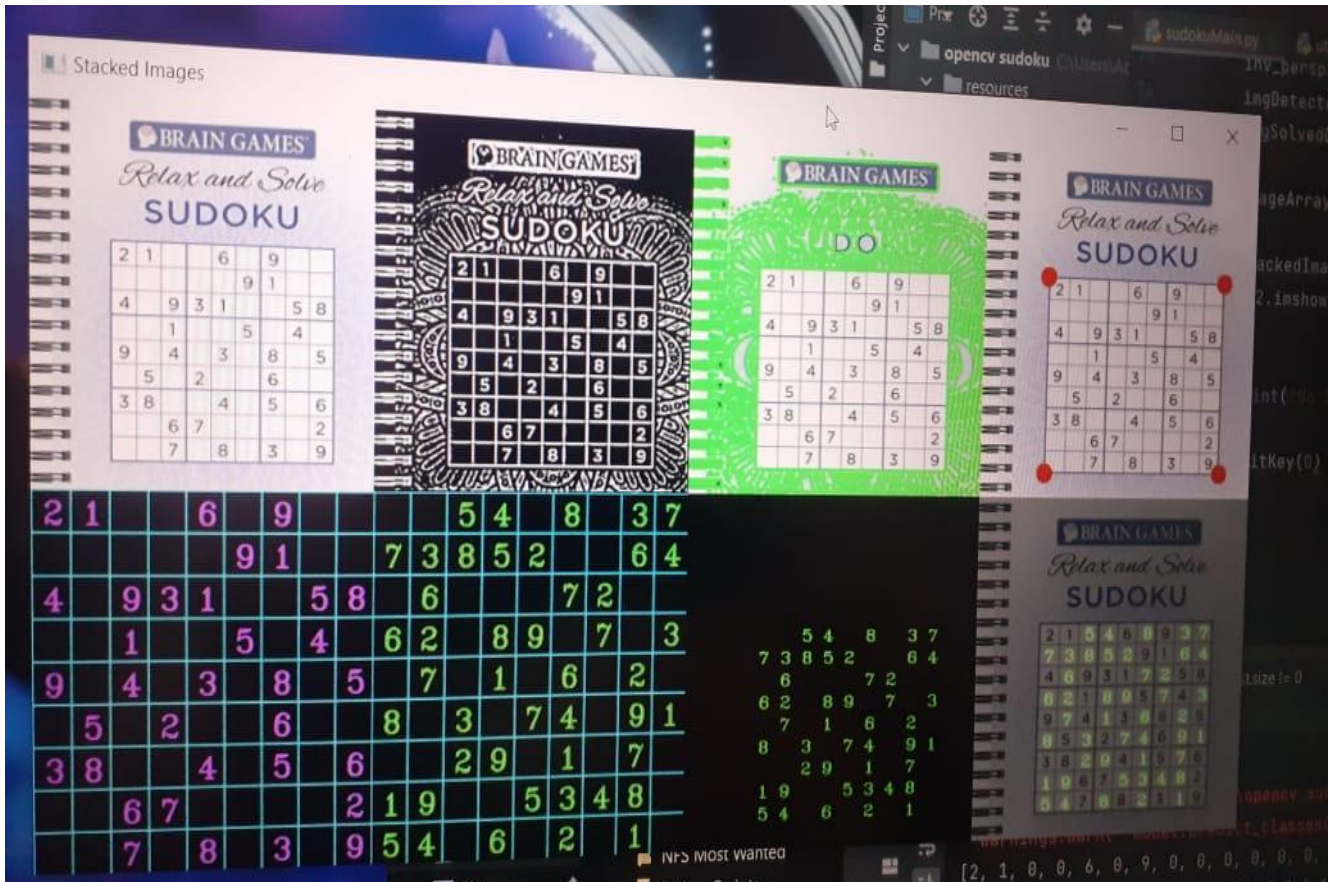
```
imgDetectedDigits = drawGrid(imgDetectedDigits)
```

```
imgSolvedDigits = drawGrid(imgSolvedDigits)
```

Feasibility Analysis

First thing is to get image. Then, at that point remove the forms, attempt to discover a sudoku board. In the event that a certified square form is discovered, project it as an orthophoto, and do some pre-handling to it. After effectively extricate the board, run Hough line change on it. In the case of everything goes right, successfully every cell is resolved. Then, at that point we can do the digit acknowledgment on every cell. Void cells are meant as 0. When we have a lattice of digits, we can utilize the solver to settle it! In this undertaking, a Z3 solver is utilized. Essentially model the issue as a CSP, then, at that point Z3 can give the plan proficiently.

- Object recognition
- Image processing
- Detecting the grid
- Separating squares and cropping digits
- Digit recognition
- Solving the puzzle



Conclusion And Future Work :

- OpenCV is the best free gadget for character affirmation yet it is outstandingly sensitive to the disturbance.
- Characters misread likewise create blunders in the arrangement of Sudoku
- We make a capacity to discover precisely if each crate was at

that point full or we need to fill it.

- We built up a framework to deal with numbers and position individually on the grounds that help to find all the crates on the matrix.
- The proposed estimation works splendidly for both: turn and planar removing inside the picture got by the camera yet isn't proposed for cases in which there is perspective, can raise this as additional reproducing work.
- At Fig. 6 shows the estimation objective, the numbers are printed upward considering the way that are not intended to turn with the image, are masterminded to follow the right position to be printed however not for have a point of view.
- The execution period of the estimation is influenced by the period of numerical objective, which hence depends upon how well they have been examined the numbers present in the Sudoku, if all stages are well the settling is appeared progressively
- we have predict the digit with the cnn model with the help of keras in this code and use open CV in processing , displaying a thresholding and making etc.

we find the correct answer with this CNN model

we extracted the digit from image and stored in numpy array and solve it with sudoku solver function.

The motive of this project has been to model Sudoku to evaluate the correlation between the symmetry and time required to solve the puzzle

True to form our test results show that balance for sure affects the arrangement time.

REFERENCES :

[1] Higashida, H.: The role of computers in puzzle world. In: Second International Conference on Culture and Computing (Culture Computing), pp. 141–142, 20–22 Oct 2011

[2] Maji, A.K., Pal, R.K.: Sudoku solver using minigrid based backtracking. In: 2014 IEEE International Advance Computing Conference, 2014

[3]<http://docs.opencv.org/modules/core/doc/intro.html>

[4]<http://code.google.com/p/tesseract-ocr/>

[5] <http://www.geeksforgeeks.org/backtracking-set-7- sudoku/>