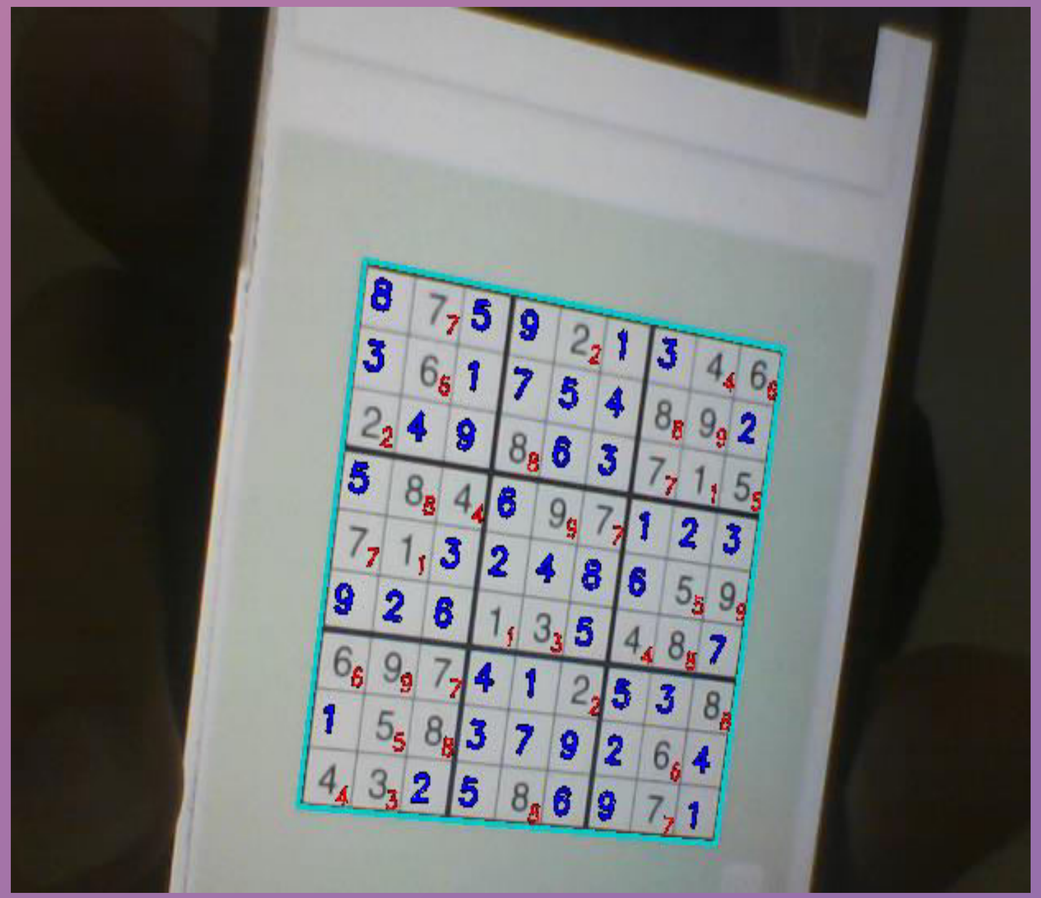
AR Sudoku Solver



About this project

The project is motivated from the concept of Augmented Reality in which objects residing in real world are enhanced by computer-generated perceptual information, which includes visual aid. In this project, we have tried to create an AR aided Sudoku Solver which shows us the potency AR has to solve so many real world problems as well. This project helped us to learn about AR that could be applied to various other fields.

Societal Impact of Augmented Reality

According to Georgia Institute of Technology professor Sterner," AR can be used to read the affect on people's faces, which is helpful for those with autism spectrum disorder who may struggle with reading social cues." Another example he gave was of a fireman who could say," Glass Show Extraction Diagram for a Ford Expedition", and it would show him how to safely extract a person from a burning vehicle.

Technical Feasibilty:

The prerequisite needed for accomplishing the implementation of this project includes:

1. Python package installed
2. Open CV installed
3. Computer Vision application
4. Augmented Reality Basics

Internet connectivity required once deployed.

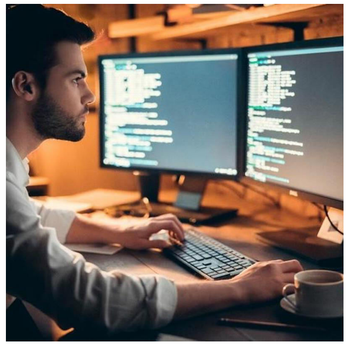
Economic Feasibility:

Economic Feasibility:

It does not require much costs in development. Only small cost required for safe deployment.

Further Scope of Project

- We can rewrite this program for Android and integrate this as an Android app and publish it on playstore.
- We can parallelize some functions to use multicore processors to improve performance. The idea is that parallel tasks should perform same operation on same image frame but with different settings.
- We can improve our backtracking algorithm from exponential time to a lot better.



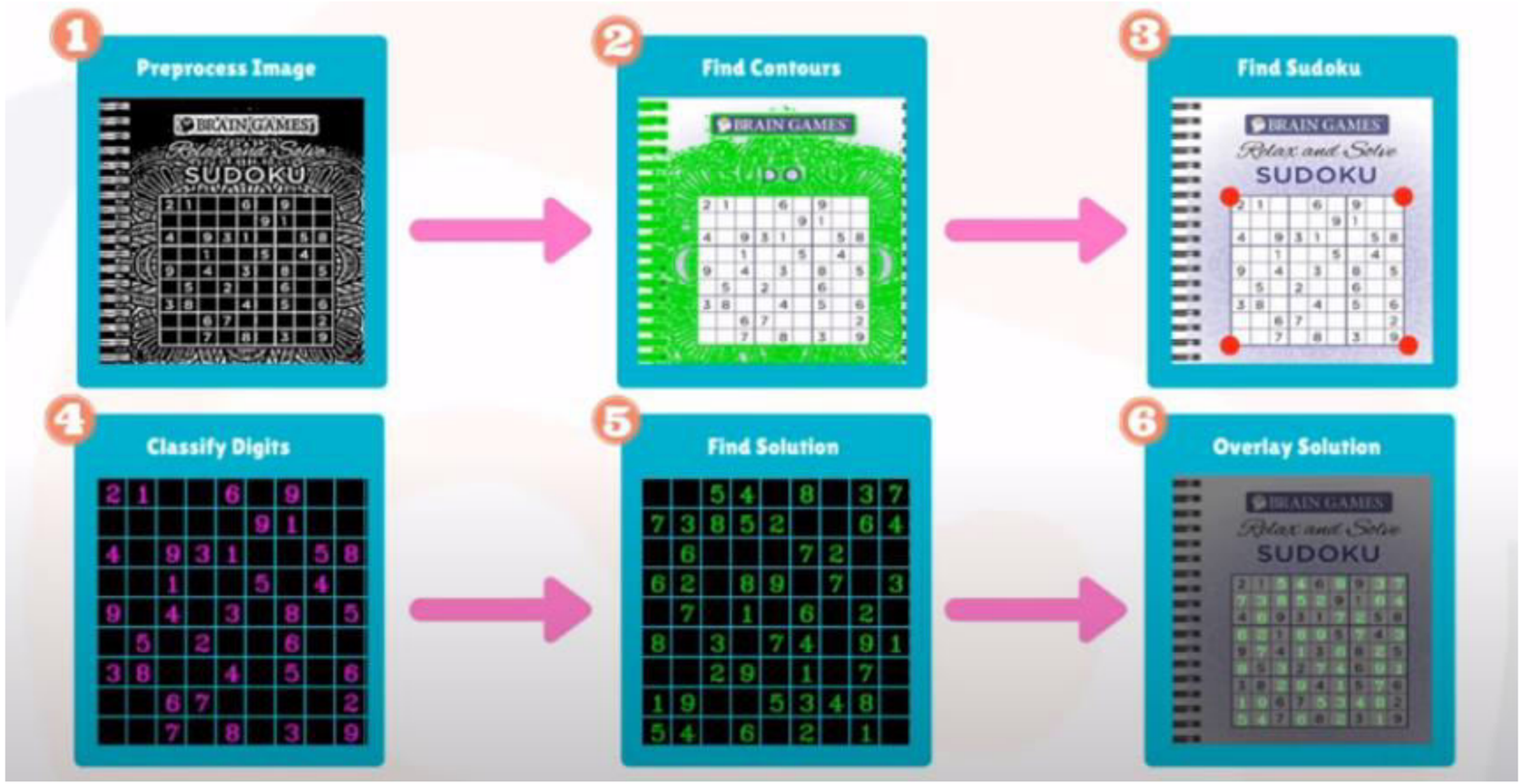
Identifying Stakeholders

a) **Developers**: The admin of our application, the one who coded and the members of the team have authorization.

a) **Students**: The one who shall use the facilities being provided by the application.

a) **Researchers:** Those who want to use it as a reference and explore the field more.

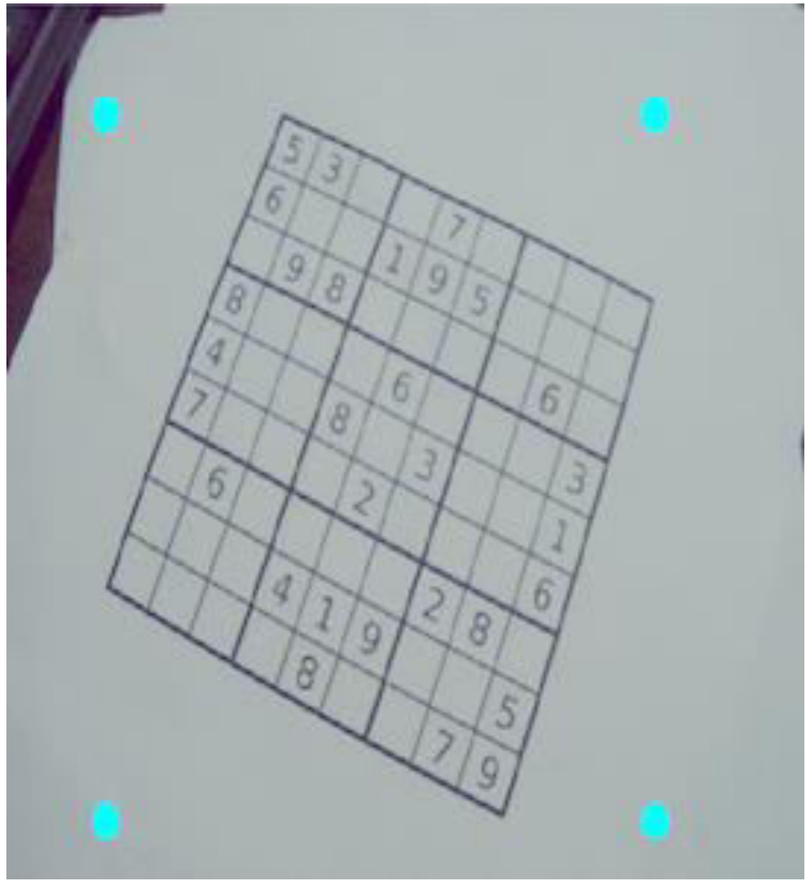
Let us look at the implementation first…

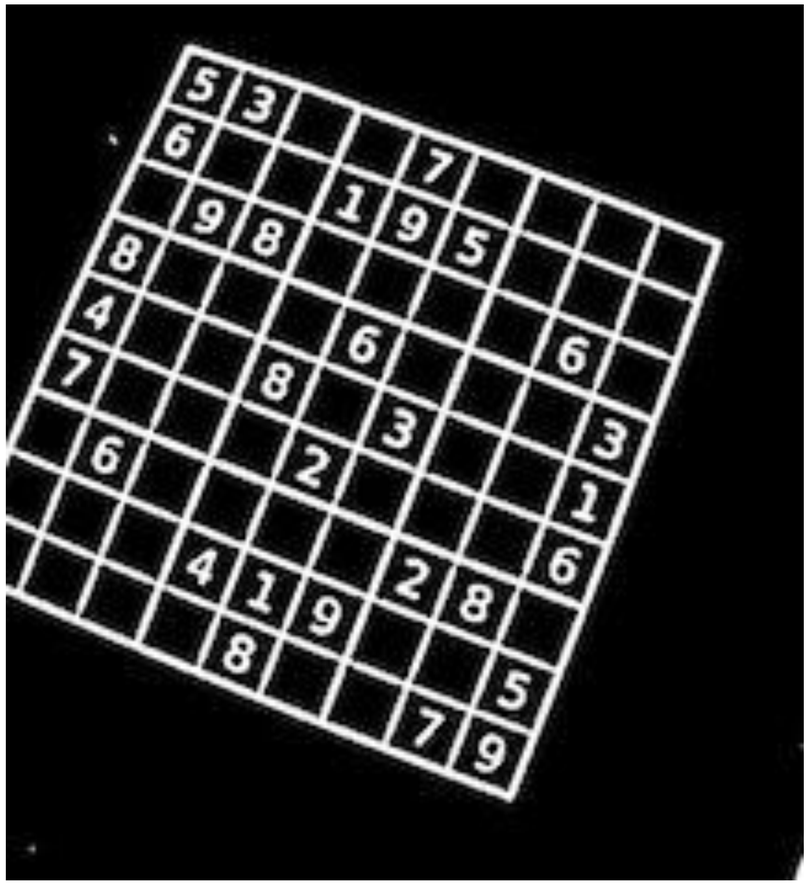


DETAILED SOLUTION

Capture Image

First step is to capture image through the webcam by the model. Here we have shown the input image that model sees.



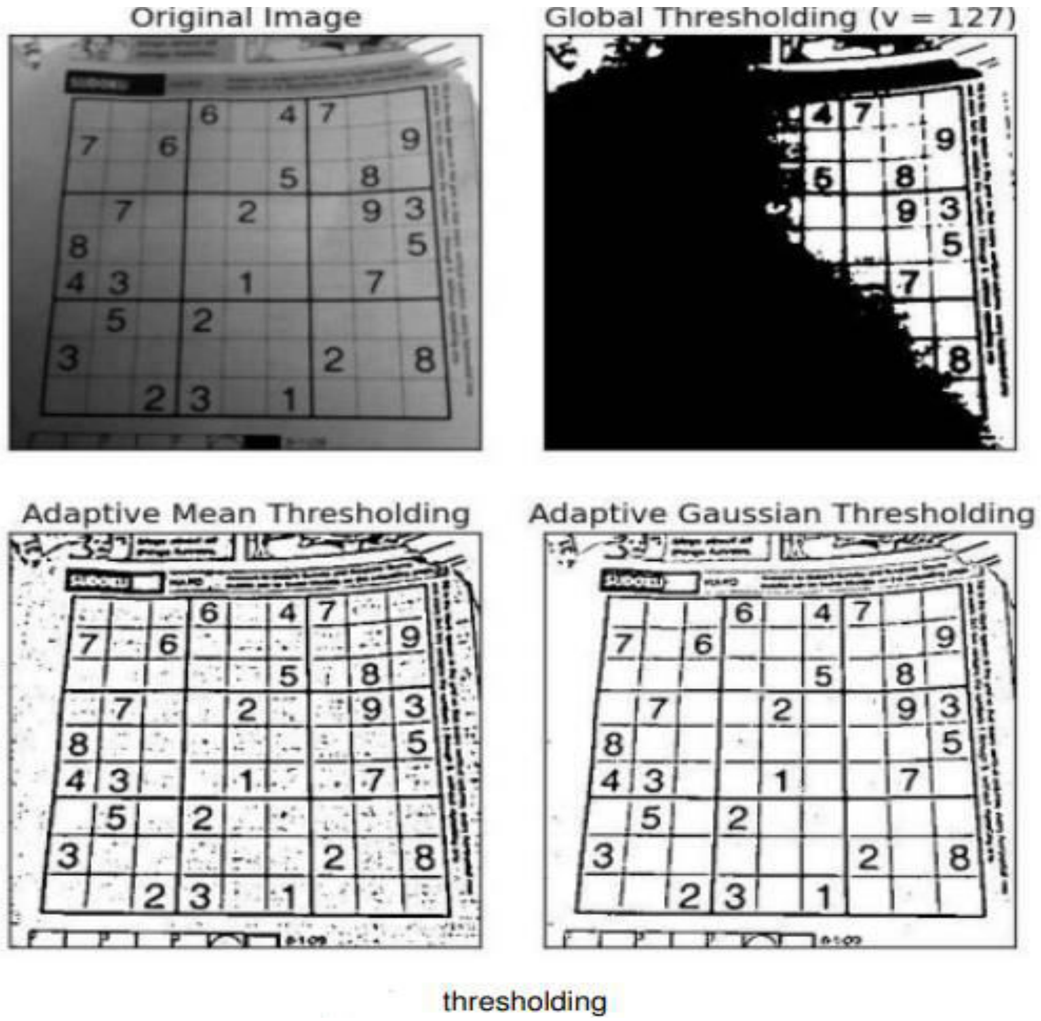


Convert to Gray Scale and Resize

Any colour present in the image to be processed does not contribute any useful information, it can be discarded by converting the image to gray scale. To store a gray scale image, only one channel is used for each pixel, compared to three channels(Red, Green, Blue) used in RGB, cutting the file size to a third of the original.
As a result, processing of the image is simpler and faster

Apply Thresholding

Thresholding is a process used to transform the gray scale image, so that only two colours exist in the image, black and white. The process is commonly used when recognizing Sudoku as it is an effective tool to distinguish an object from the background. Global thresholding is not very effective in cases where different lighting conditions are present in the image, such as glare. Adaptive thresholding is a similar process to global thresholding that tackles differences in spatial illumination across the image. This is achieved by partitioning the image to various ”neighbourhoods” and calculating a different threshold value for each.



thresholding

Apply Gaussian blur

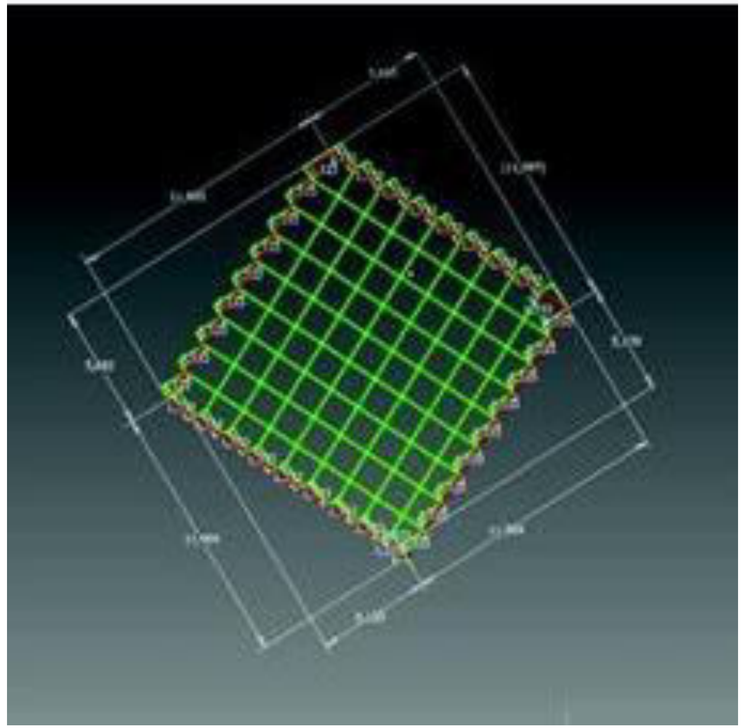
Gaussian blur is a technique used to smooth out any noise that could be introduced
during the camera's capturing process. It is based on the Gaussian function:

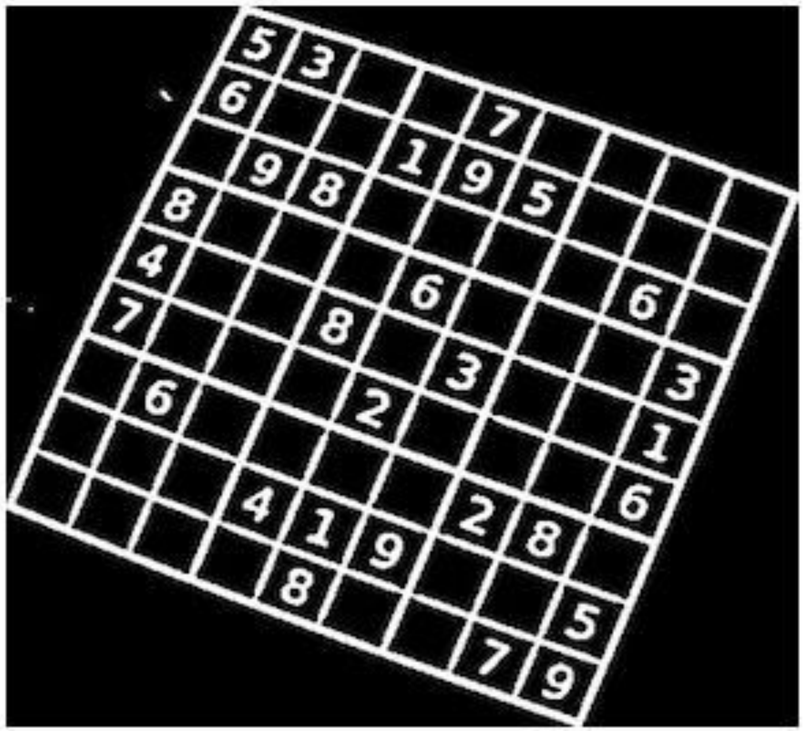
$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$$

Drawing a Grid

We draw the grid for two reasons:

1. To identify the correct location of the boxes to determine whether it has a number or not
2. The second reason is because it allows us to give an augmented reality on our Sudoku.





Largest rectangle (Rotated) contains sudoku puzzle.

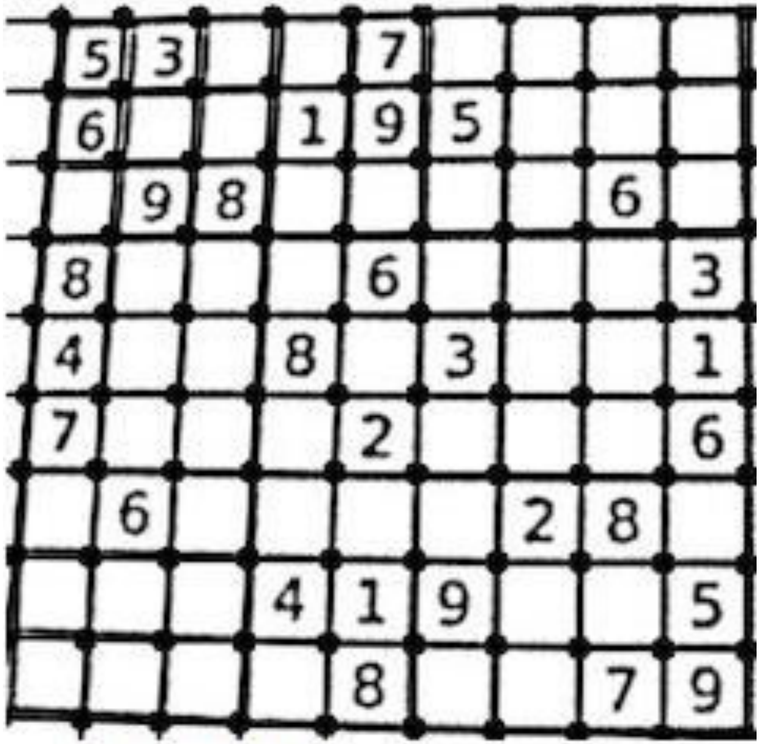
Finding contours and best contours

Now we can use OpenCV's findContours algorithm to detect shapes in the image; the RETR_EXTERNAL mode is used to limit the search to only extreme outer contours skipping any that lie inside other contours.

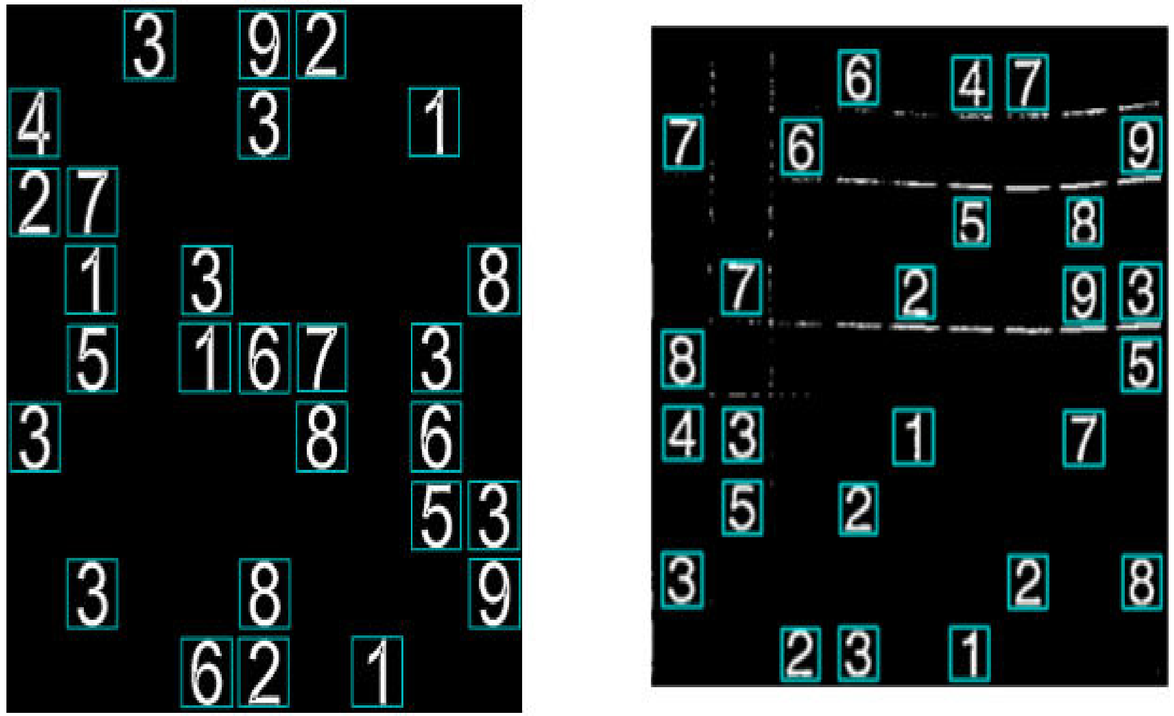
With an array of contours, which are simply lists of X/Y coordinates that "trace" the shape, the next task was to find the largest contour with the help of contourArea and then identify which X/Y coordinates are the corners of the grid by finding the outermost point in the contour within each quadrant of the area covered by the contour.

Applying Warp Perspective

These corners are then used to “un-warp” the grid section of the image with OpenCV’s getPerspectiveTransform and warpPerspective. This basically means that we have cut the sudoku out from the rest of the background.



After warping



Detecting Digit Regions of the Image

Finding digits and separating them

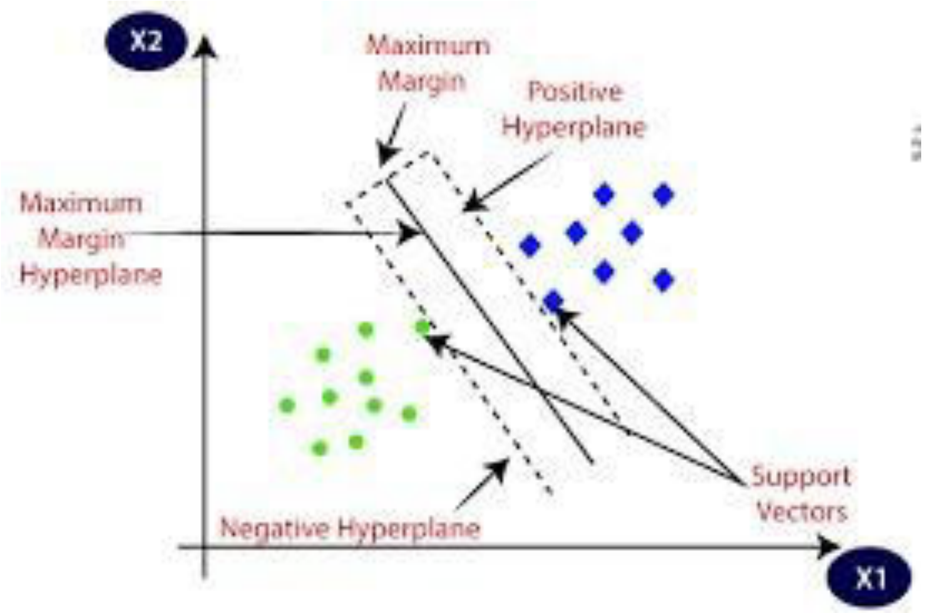
Finally it's time to detect the digit regions of the image. Once again findContours() is used followed by another aspect-ratio filter. This filter allows us to skip over contours for some of the horizontal artifacts left in the newspaper image on the right.

Extracted cell

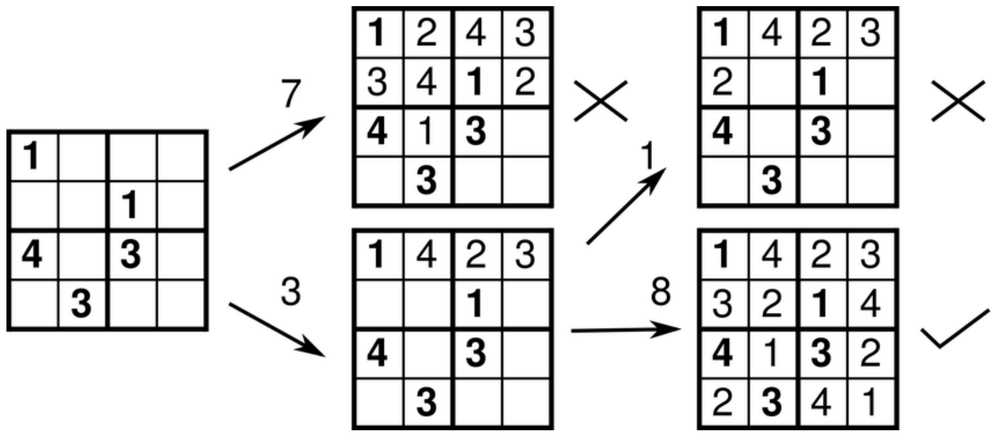
Extracted digit

Classify the digit using SVM

We have used the in built SVM model of open-cv with a linear kernel & trained the model for
1000 epochs. The model was trained on a labeled training data which included images of
digits from 0 to 9 & were labeled by hand. Training accuracy obtained was 100%. We used SVM because if we would have used CNN, our model would not be a light weight algorithm and would occupy a lot memory on deployment.



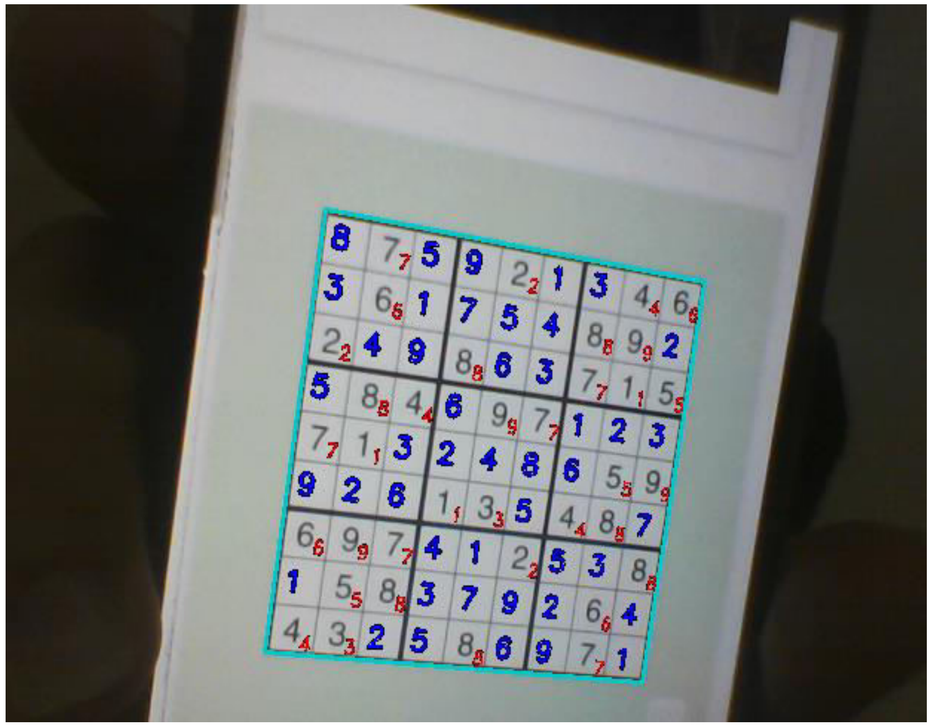
Finally solve the Sudoku using Backtracking



Finally it is time to solve the obtained digital Sudoku using Best First Search Algorithm (Backtracking) algorithm which is very simple to implement but takes exponential time of the order of 9^m where m is number of blank spaces in the Sudoku.

Finally print the solution

In this step, we print the result to the image, the blue digits are the solution, and the red digits
are the recognized digits of the Sudoku puzzle. The green boundary is the contour of the Sudoku puzzle.



UML Diagrams

- The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. UML diagrams helps us to understand the flow of application, how the actual mechanism works and what type of functions are available in the project.

- UML diagram is basically of two types:
  Structural Diagrams: Structure diagrams show the things in the modeled system. In a more technical term, they show different objects in a system. Eg: class based diagrams
  Behaviour Diagrams: Behavioral diagrams show what should happen in a system. They describe how the objects interact with each other to create a functioning system. Eg: use case diagrams and sequence diagrams.

In our project, we don’t have class based implementations rather we have some functions that help in achieving the end product. So we have made Use case diagrams and Sequence diagrams.

MiConcept

Use Case Diagram

In our project, we have two actors (user, programs) and between them only our whole application runs. There is only one major function involved in our program i.e. solving Sudoku by presenting a picture of unsolved Sudoku to the camera of the device.

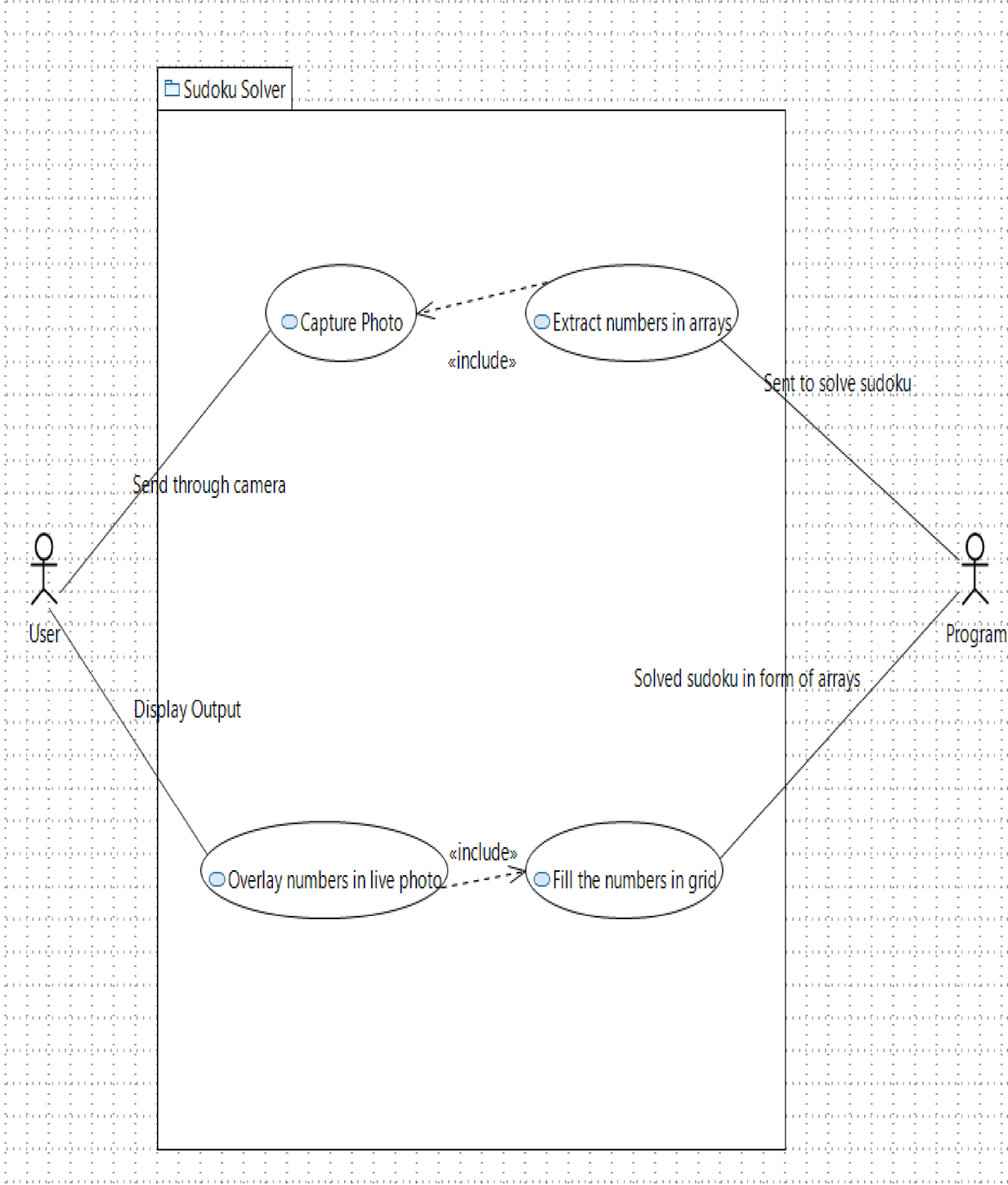
Functions included in our program are as follows:
a. Device's camera captures photo

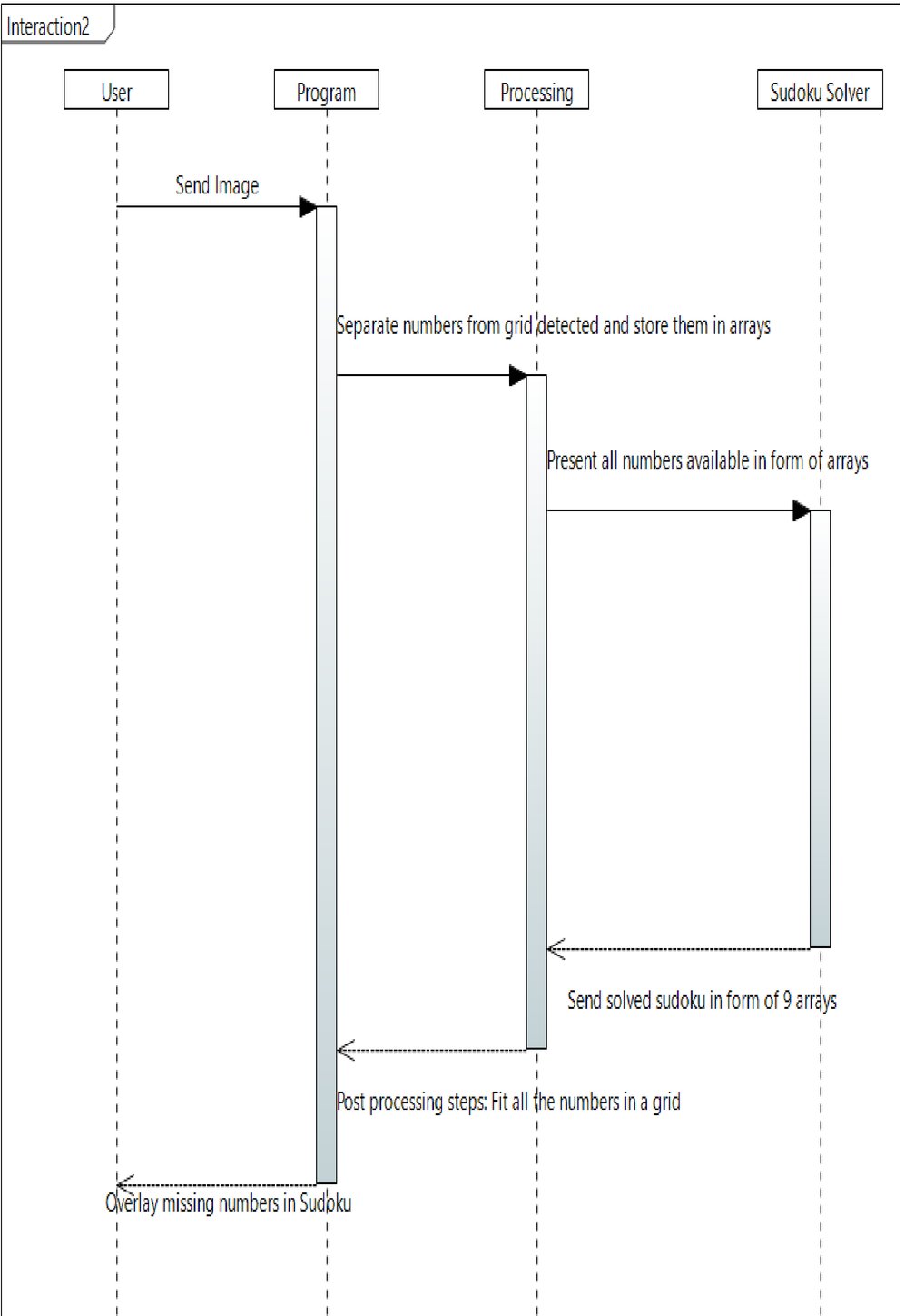
b. Numbers are extracted from the photo captured.

c. Numbers are sent in form of arrays to the program to solve Sudoku.

d. Program sends solved Sudoku in form of arrays

e. arrays are converted into grid and missing numbers are overlayed on the unsolved Sudoku grid.





Sequence Diagram

For our case we have the following sequence diagram:

a. First the user interacts with the main program by sending the image of the unsolved
Sudoku.

b. Program processes the image by identifying the contours of the grid. Then the numbers identified are separated from the grid and stored in form of arrays. Now these arrays are passed to Sudoku solver program.

c. Sudoku Solver program then solve the Sudoku using backtracking algorithm and send the solution back to processing unit in form of arrays.

d. Now the arrays are converted into a grid 9*9. This grid contains the whole solution of the
given Sudoku problem.

e. Finally the missing numbers are overlayed on the live image of unsolved Sudoku.

Tech Stack Analysis

In order to achieve proper solution, we have used a variety of Tech Stacks. All these technologies have been chosen on the basis of the following few criteria:-

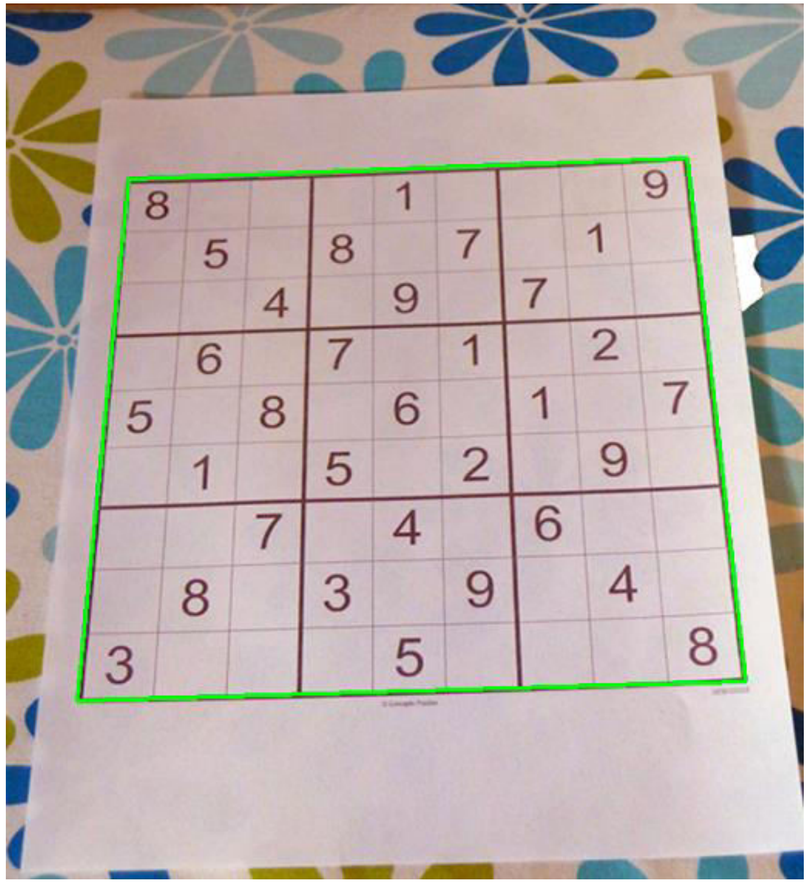
1. Ease of Usage and Ease of Learning

2. Efficiency

3. Accuracy

OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. OpenCV is written in C++ and its primary interface is in C++. There are bindings in Python, Java and MATLAB/OCTAVE for using this library.



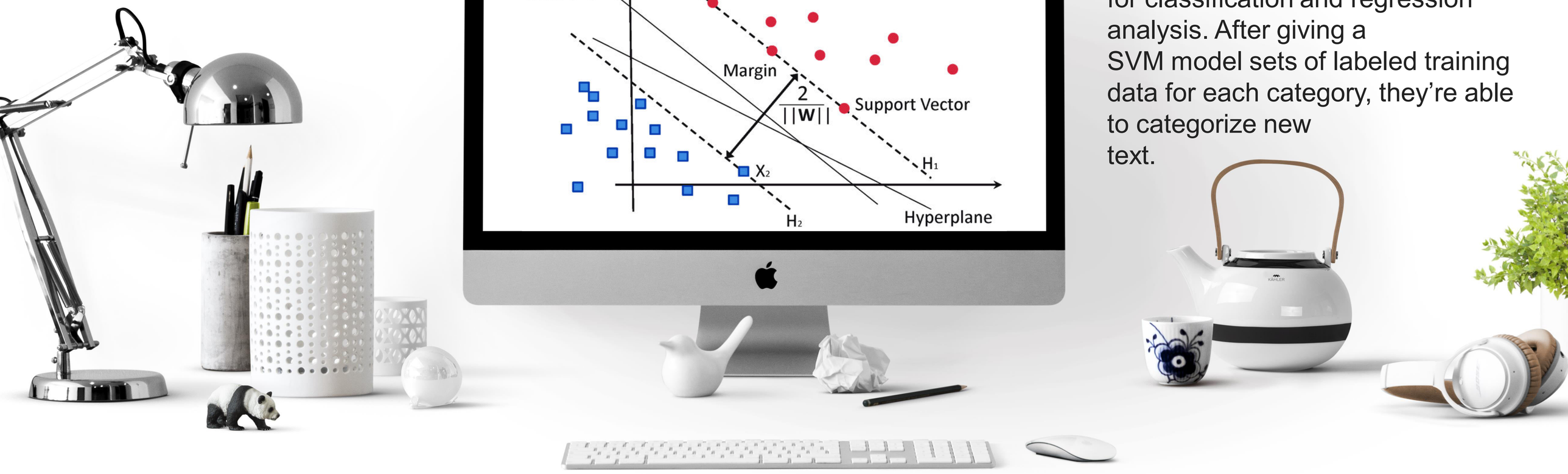
Use of OpenCV in our project

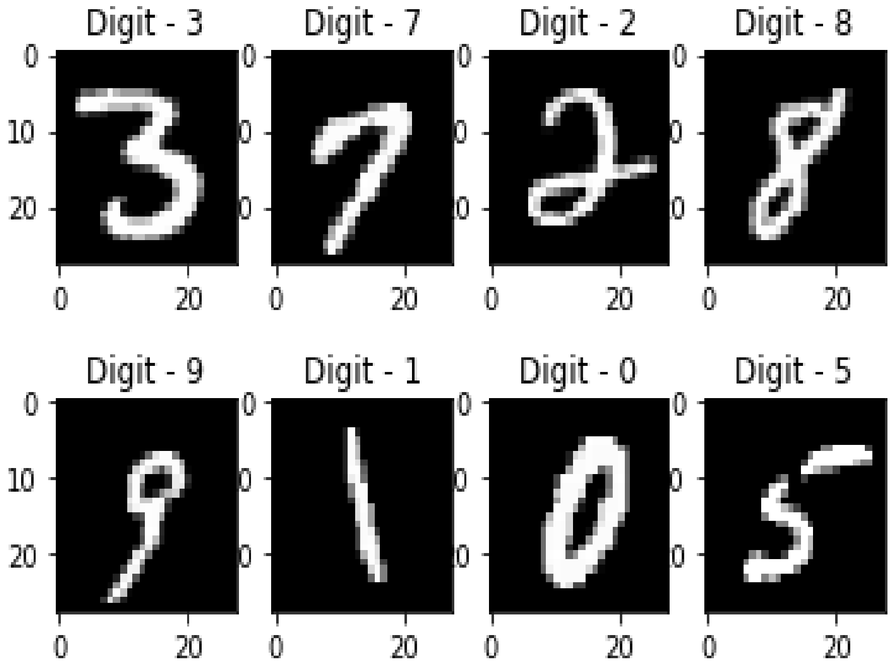
OpenCV helps to capture image of unsolved Sudoku from user's device.
When a user starts a program he/she positions the Sudoku in front of camera in such a way
that program is able to capture the grid of Sudoku.

MiConcept

Support Vector Machine

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. After giving a SVM model sets of labeled training data for each category, they're able to categorize new text.





Use of SVM in our project

Main use of SVM in our project is to identify the digits in a Sudoku grid. After the grid is identified the numbers are classified according to the model we trained on SVM. These identified numbers are then sent to Sudoku solver program for final output.

MiConcept

Heroku

Heroku is a cloud platform as a service supporting several programming languages.
Heroku is a platform where we can deploy our application freely on the internet. It involves all the facilities like cloud database and assigning domain names to the applications.

Usage Instructions

This python program is very easy to use. User has to follow these set of instructions:

a. Position the camera in such a way that the grid of Sudoku comes exactly in the visible range of camera.

b. He/she should avoid bringing anything else closer to the camera so that camera can focus on the grid only.

c. He/she should check whether the device camera is clean, if not then he should clean the camera first so that image can be clearly captured.

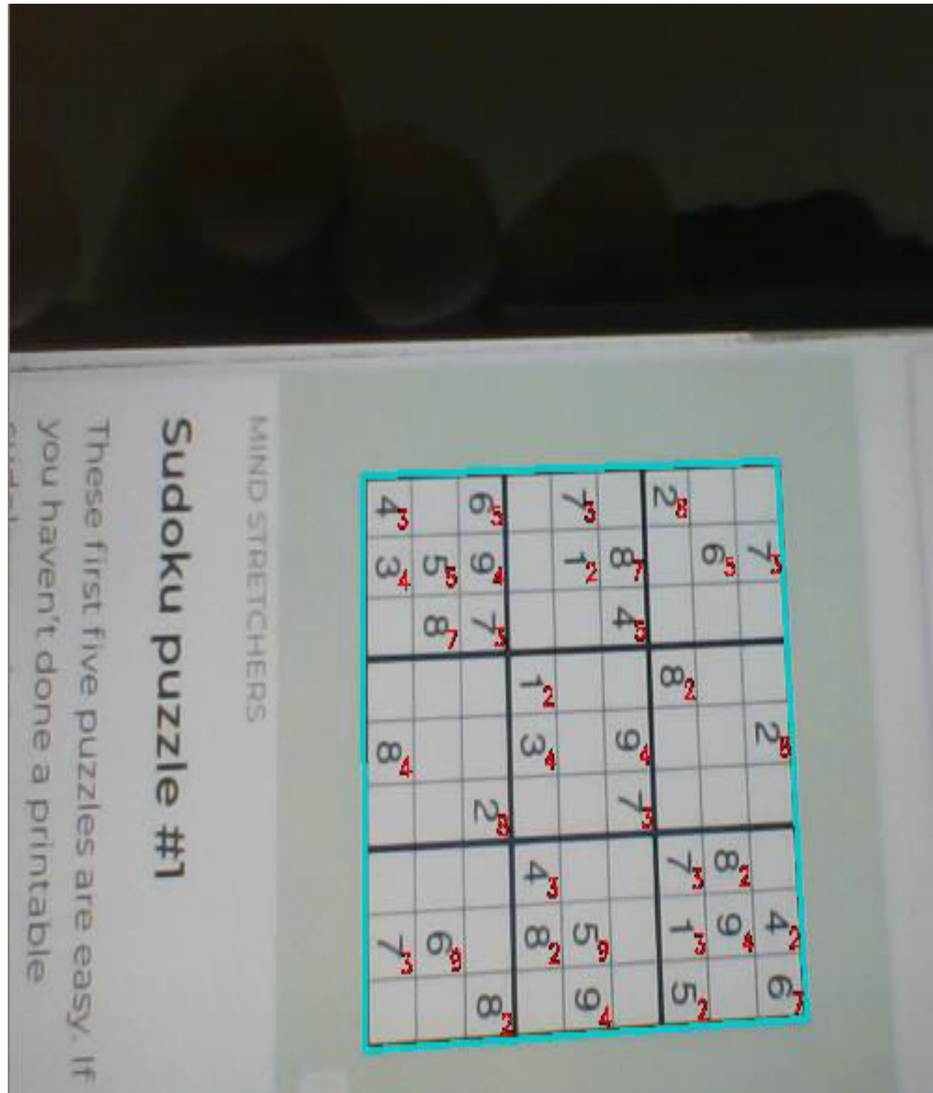
d. User should keep the unsolved Sudoku still in front of camera to avoid blurry image.

e. User must keep Sudoku properly aligned in front of camera.

f. User must keep the hand holding Sudoku steady till the missing numbers have been overlayed.

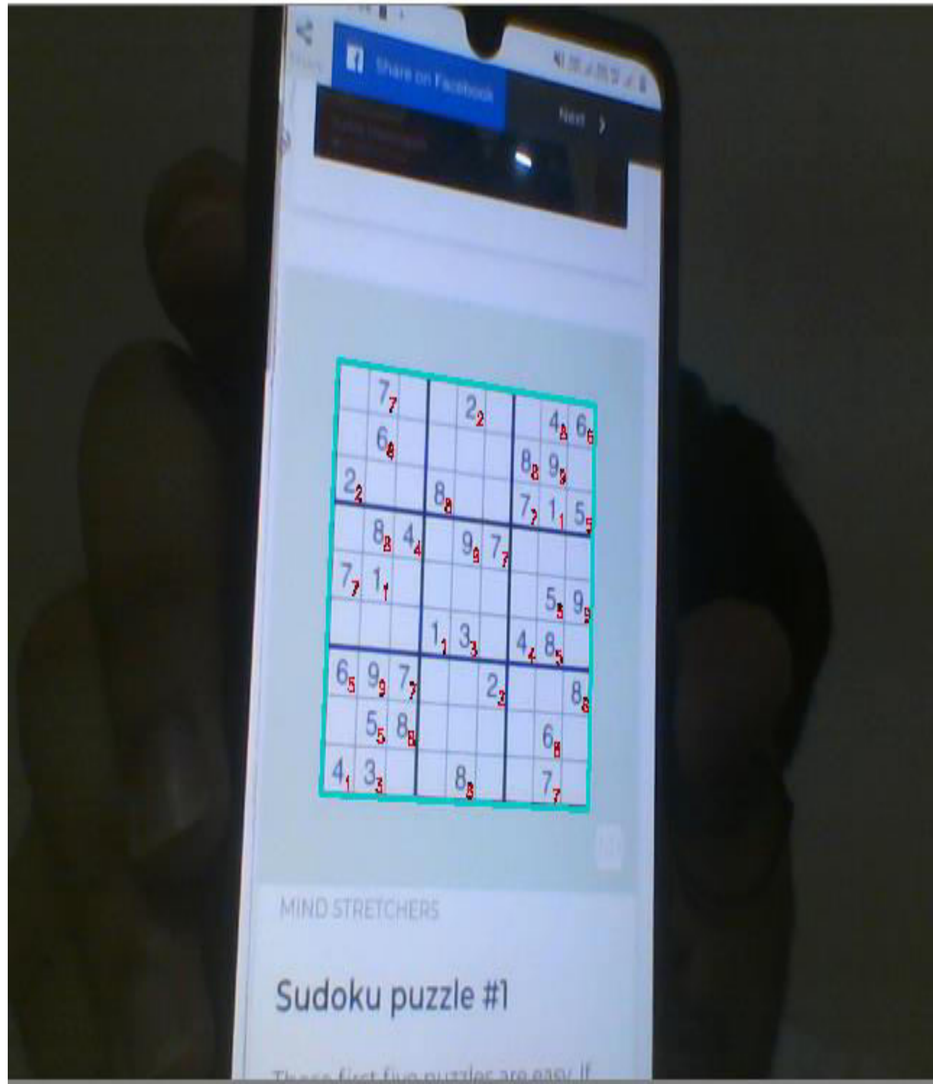
Wrong way of holding Sudoku

We can see that holding the Sudoku in horizontal direction leads to wrong classification of numbers. This is because due to time constraint we were not able to add rotation skew feature in this model.



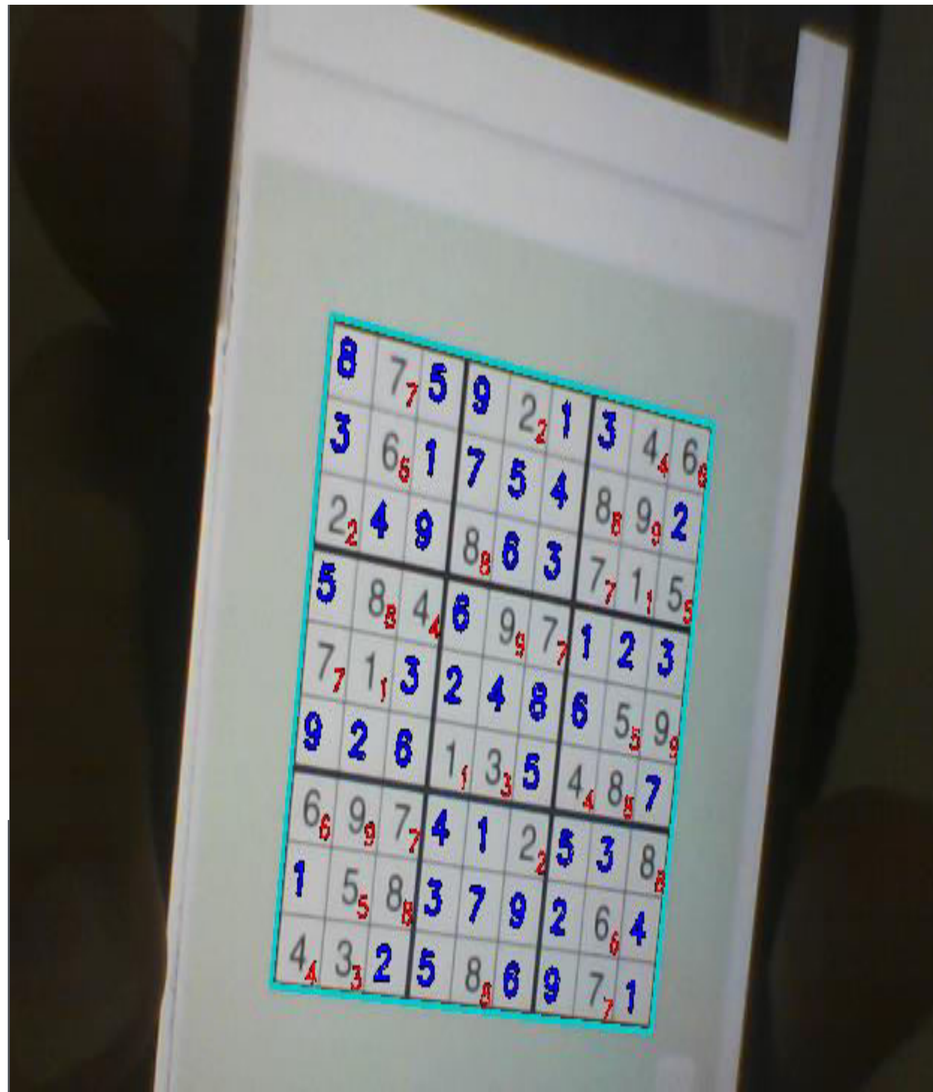
Correct way of holding Sudoku

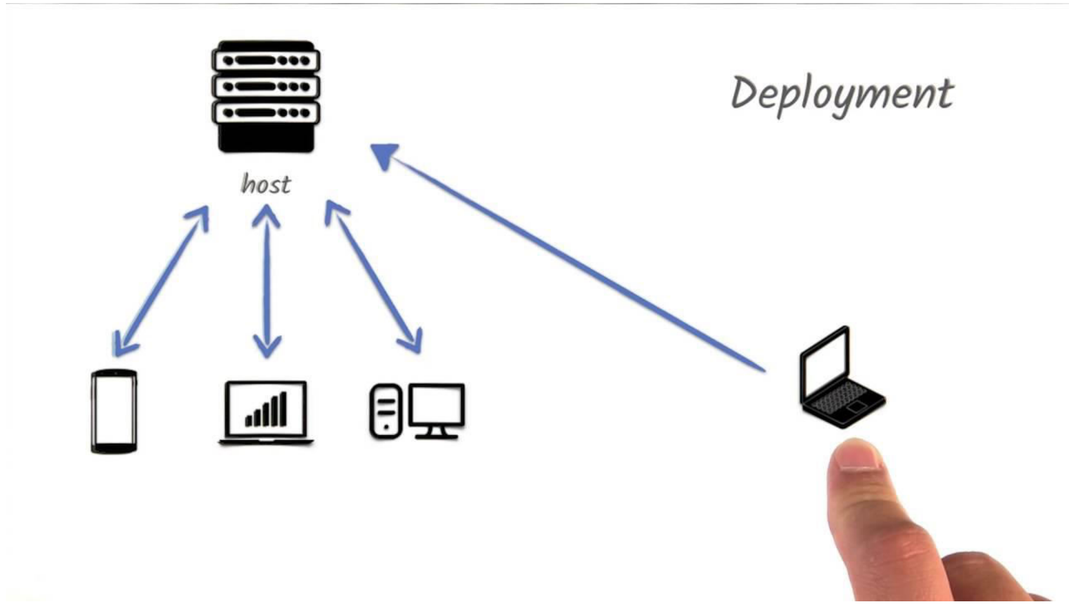
This is the correct way of holding Sudoku where the Sudoku is held almost vertical in front of the web cam with very little amount of tolerance for rotation of camera.



Augmented Reality aided Sudoku solver

On holding the Sudoku in correct manner, we observe that the solution is printed back in the grid with blue numbers as the solution and red numbers as the original digits. We observe the solution after few seconds due to the exponential time taken by Sudoku solver backtracking algorithm.





Deployment status

This program will be deployed on internet using Heroku application in the coming future.
Testing will also be done after deployment. We were not able to do it now due to time constraints.

Conclusion

The role of computers in puzzle world is now becoming more and more important because
computers changed not only puzzle creators, but also puzzle solvers.

Some future work references are-

• The solver is by no means fullproof, it still has trouble with some images and either will fail to parse them or parse them incorrectly leading to failure to solve them.

• The execution time of the algorithm is influenced by the stage of numerical resolution, which in turn depends on how well they have been read the numbers present in the Sudoku.

• It also uses backtracking algorithm that runs in exponential time complexity.

This project is further used for other Augmented Reality related operations which can have a
major impact on society.