An-Najah National University

Faculty of Engineering and Information Technology

Computer Engineering Department

Graduation Project II

“Sudoku Solver”

Prepared by:

Rami Abu Seir & Khaled Abulibdeh

Supervised by:

Dr. Raed Qadi

Presented in partial fulfillment of the requirements for bachelor’s degree in computer engineering.

Acknowledgment

First and foremost, we give thanks to God for blessing us and for using us to carry out this endeavor. We also pray that the knowledge we gain will be useful to the rest of the world, especially in Palestine.
We would like to thank our families for the continuous support and encouragement they provided us from the beginning until the last moment.
We thank Dr. Raed Qadi for his constant help and for providing us with a lot to complete this project.
We thank our dear Drs in the Department of Computer Engineering, who had the credit for us by reaching this stage with all that they have given us throughout these years, we will forever be grateful.
To all those and many more, who supported us through thick and thin, we give a big “Thank You” straight from our hearts, and we present you this report and project.

Disclaimer

This report was written by Rami Abu Seir and Khaled Abulideh at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, because of the assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was Commissioned.

Table of contents

Table of Figures

Abstract

Sudoku Solving Robot is a robot that can help players solve and fill the sudoku grid automatically, making it easier to solve for players learning the game or stuck in the process of solving. This is achieved by using image processing and machine learning, combined with solving algorithms and puzzle generating algorithms to provide the user with the best experience.

Features included in the project:

1. Provide an Android mobile application to select modes.
2. Provide a mode to solve a provided puzzle.
3. Provide a mode to print a puzzle.
4. Provide multiple difficulties for requesting puzzles.
5. The ability to give the puzzle in handwriting and print.
6. Using a camera to scan the grid on a paper.
7. Using a CNC machine with a pen to fill the results on the paper in solve mode, and to print the puzzle in puzzle mode.

Report Organization

- First chapter: In this chapter, we will go through the introduction of the project including the problem statement, objectives, scope and significance of the project.
- Second chapter: We will list the constraints we faced and mention earlier coursework that proved to be helpful during the work on this project.
- Third chapter: We will briefly discuss the areas our project covers and some information about how these areas relate to the project in other papers.
- Fourth chapter: In this chapter, we will go into details of how the project works including all the algorithms and hardware parts and tools we used.
- Fifth and final chapter: We will show the contribution of this project and discuss the results

Chapter 1: Introduction

1.1: Statement of the problem

When learning a new skill, it is very common that someone can find difficulty in the process. And when tackling obstacles like these on the way, frustration is extremely bound to happen.
Using Tools to help overcome these issues is a proficient way of learning and getting better. This project comes as a solution tool for dealing with Sudoku puzzles, whether the user is a beginner who is learning the puzzle, or an experienced individual who is finding difficulty in solving a puzzle, or even a person who is looking for a way to get puzzles.

1.2: Objectives and Scope

As said in the previous subsection, the purpose of this project is to help individuals with solving and learning the Sudoku puzzle.
The objective that this project seeks to achieve are:

1. Giving users an easy way of dealing with Sudoku puzzles with the provided mobile application.
2. Providing the user with a functional way of seeing results on paper.
3. Providing the user with different puzzle difficulties to accommodate their experience levels.

Chapter 2. Constraints and Earlier Coursework

2.1: Constraints

2.1.1 Inaccuracy of machine learning models

Since we are reading the values from the paper, we needed a machine learning model to recognize the digits written in the boxes. This model proved to be inaccurate in reality even though it passed with a high accuracy percentage during the making of the model.

2.1.2 The lack of funds and high cost of parts

To produce an accurate output, we needed extremely accurate motors which proved to be very expensive. So, we ended up with less accurate models of motors which contributed to the inaccuracy of the CNC machine and therefore the accuracy of printing.

2.1.3 Lack of resources

During our work, we found the need to look for external libraries and scripts to achieve the results we need, we found a lack of libraries for certain tasks like dealing with images and GCODEs for the CNC machine. Which inevitably lead to more time wasted trying to find workarounds for these tasks.

2.1.4 Low quality of some parts

Some of the parts we used were not in the best condition because we opted to get used parts to reduce costs. Examples of these include the PiCamera which had degraded quality and colors which needed to be fixed with image processing techniques, and the stepper motors which were slow and meant that the CNC was slow and needed more time during testing.

2.1.5 Lack of technical knowledge

We needed to learn new languages to be able to make sure that the project works, like learning Kotlin to make the mobile application.

2.2: Earlier Coursework

2.2.1 Digital Image Processing

The course was a big part of the project, most of the process used to get the image from the camera and into the machine learning model was made of things we learned in the course.

2.2.2 Microcontrollers and PIC

All the basics of the Raspberry, Arduino, ESP32 like I/O and basic serial communication were taken in these courses.

2.2.3 Artificial Intelligence

The main parts of the project are the solving and generating algorithms. To solve a Sudoku puzzle we needed to use a backtracking algorithm to get to the solution.

2.2.4 Operating Systems

We used the Raspberry Pi OS, which we flashed on the Raspberry and used the Linux command line to install libraries and packages needed for the project. All those are some things discussed in the course.

2.2.5 Networks and Communication

We implemented the knowledge from this course in many aspects, like setting up an ssh connection with the Raspberry Pi to control it from a laptop, we also needed the knowledge to manage the connection between the Raspberry Pi and the Arduino and ESP32 modules using wifi protocols.

2.2.6 Critical Thinking and Research Skills

The research and the writing of this report was done using the knowledge gained from this course.

2.2.7 Electronic circuits courses

These courses highly contributed in dealing with different modules and finding the right power sources and connection for each of them.

2.2.8 Programming courses

Mainly C++ and OOP for writing the codes and algorithms.

Chapter 3. Literature review

Sudoku has always been a puzzle that interests people and excites them to think and look for solutions. Since number puzzles appeared in newspapers in the late 19th century, people started taking interest in logic-based combinational number puzzles. The Sudoku we know now developed in Japan and spread over the world fast.

There have been several past research studies that have tackled various aspects related to this project.

In terms of image processing, there have been studies that have focused on techniques for accurately recognizing and interpreting images of Sudoku puzzles. These techniques include image thresholding, edge detection, and pattern recognition. These techniques have been shown to be effective in recognizing the grid and digits in the puzzle.

In terms of puzzle solving algorithms, constraint satisfaction and search algorithms have been widely used to efficiently and accurately solve Sudoku puzzles. These algorithms have been shown to be effective in finding the solution to a puzzle and can be optimized to improve the solving speed.

In terms of puzzle generating algorithms, mathematical algorithms have been proposed to create valid and challenging Sudoku puzzles. These algorithms aim to create puzzles with specific levels of difficulty and constraints to provide a wide range of challenges for the players.

Research related to the use of mobile application, handwriting recognition and puzzle generation have been also done in the past, However, the proposed project is unique in its combination of all the mentioned features and the use of CNC machine with a pen to fill the results on paper in solve mode and to print the puzzle in puzzle mode.

Chapter 4. Methodology

This section is intended to show detailed information about the parts, methods and techniques used to develop the robot and accompanying mobile application.

4.1 Image Capturing and Digit Recognition

4.1.1 Image capturing

To get an image of the paper, we used PiCamera to capture the image and store it on the Raspberry Pi to access it and extract the information. This was done using the *libcamera* library.

Fig (1): Raspberry Pi Camera rev1.3

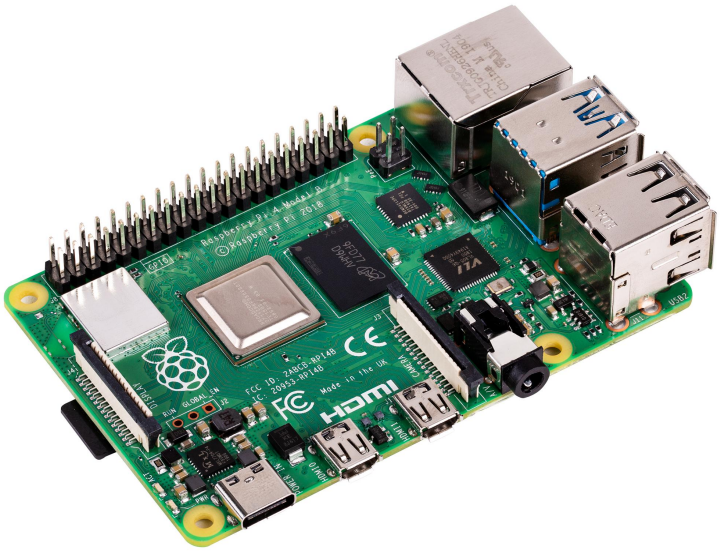


Fig (2): Raspberry Pi 4

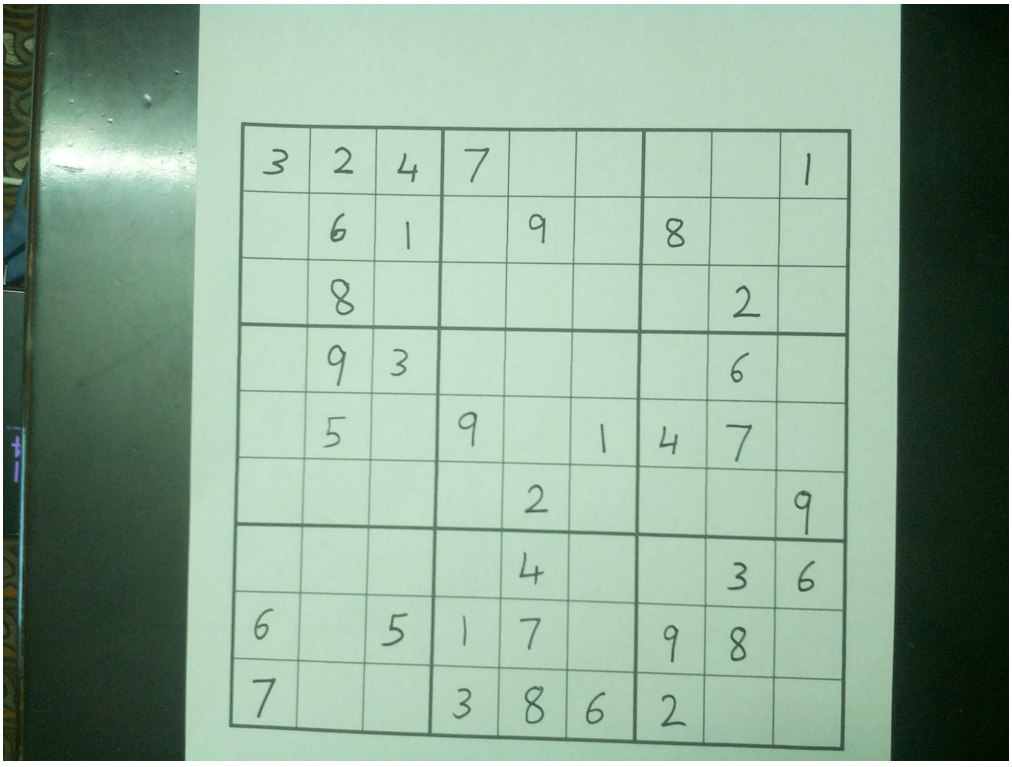


Fig (3): Image Captured by Camera

4.1.2 Image Processing

To get the individual digits from the images and model them to input for the machine learning model. Many techniques were used. We will look at them in detail:

1. **Importing the image and resizing**
   The image is imported into the python code using OpenCV and resized to be squared.
2. **Converting the image to grayscale and applying threshold**
   This step is done to denoise the image a little bit to prepare it for further processing.
3. **Contouring**
   Extracting the external contours from the image to look for the sudoku grid placed on the paper.

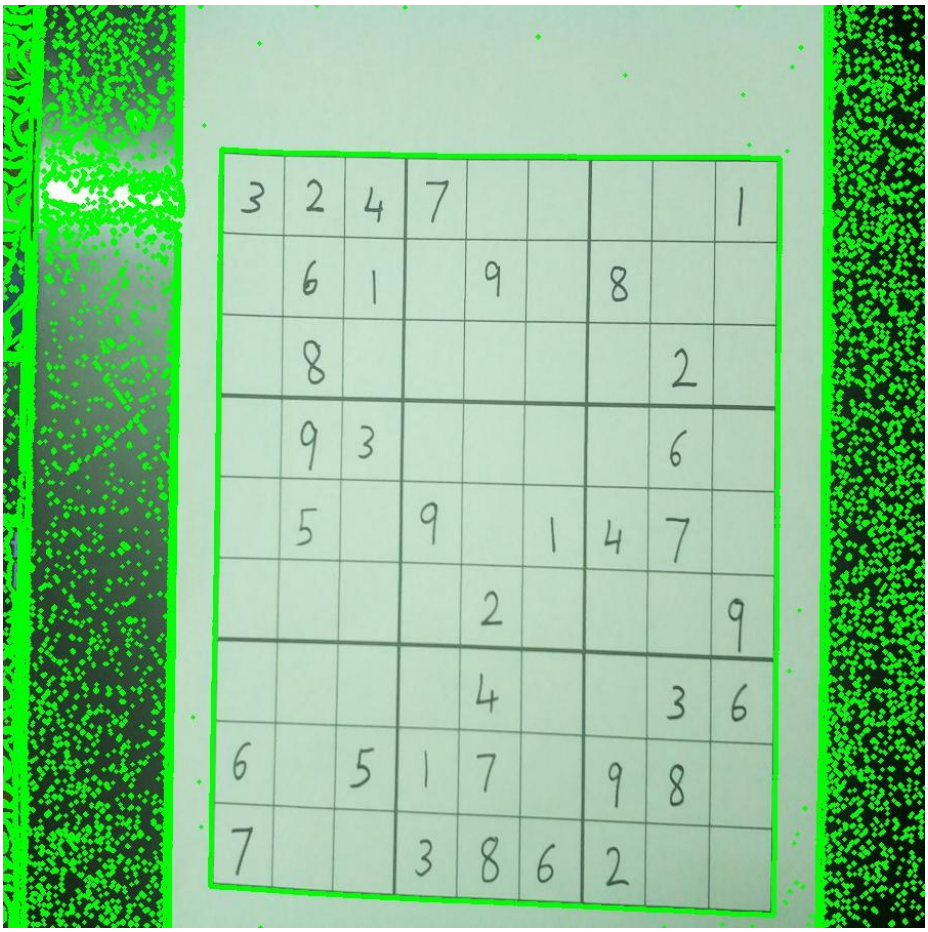


Fig (4): Contours Applied to Original Image

4. **Warping**

Looking for the biggest area of the contours yields the sudoku grid's four corners. Reordering the points correctly depending on the (x,y) values gives us the corners to provide to the warpPrespective function so we get only the square grid.

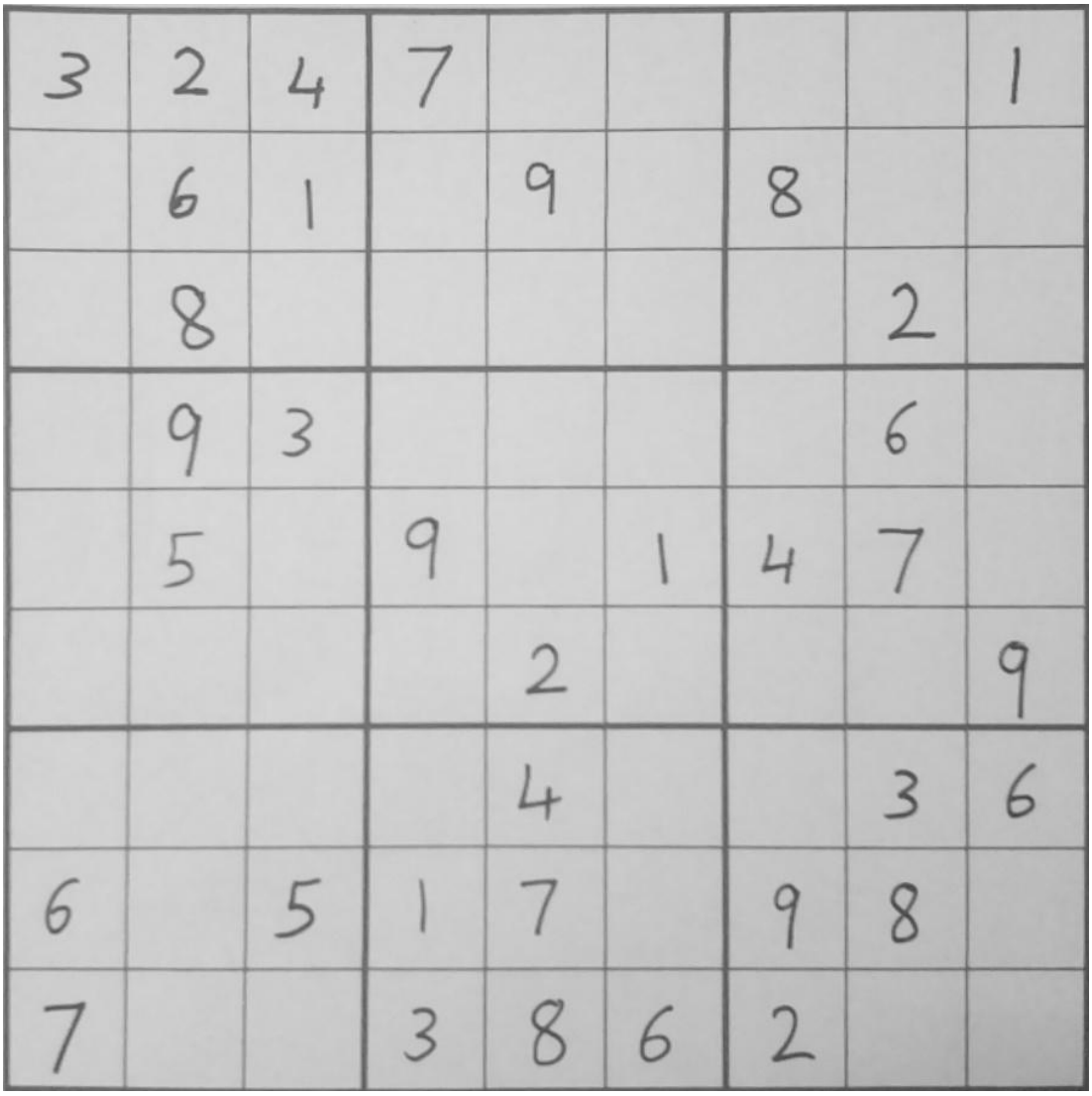


Fig (5): Warped Image of the Isolated Grid

4.1.3 Digit Recognition

4.1.3.1 Preparing Images

Now that we have the grid isolated, we can start preparing for the machine learning model. The first step is to apply thresholding to get an image with a black background and a white font.

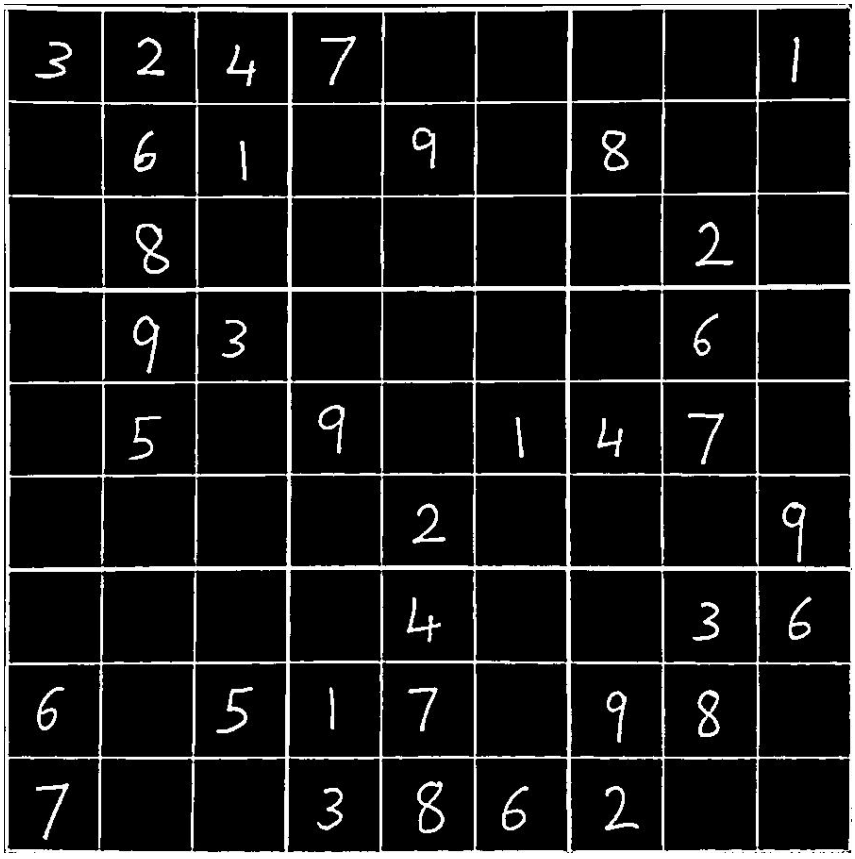


Fig (6): Thresholded Image of Grid

Next step is to split the image vertically and horizontally to get each block alone. We also apply a resizing function to match the machine learning model training data.

Fig (7): Split Images of Individual Boxes (empty and filled)

4.1.3.2 Machine learning model

For the machine learning model, we needed to get a dataset of images of numbers, we used the MNIST dataset. This dataset included 60000 training images evenly distributed between all numbers, and 10000 testing images. Using a convolutional neural network, we trained the model for 5 epochs and these were the results

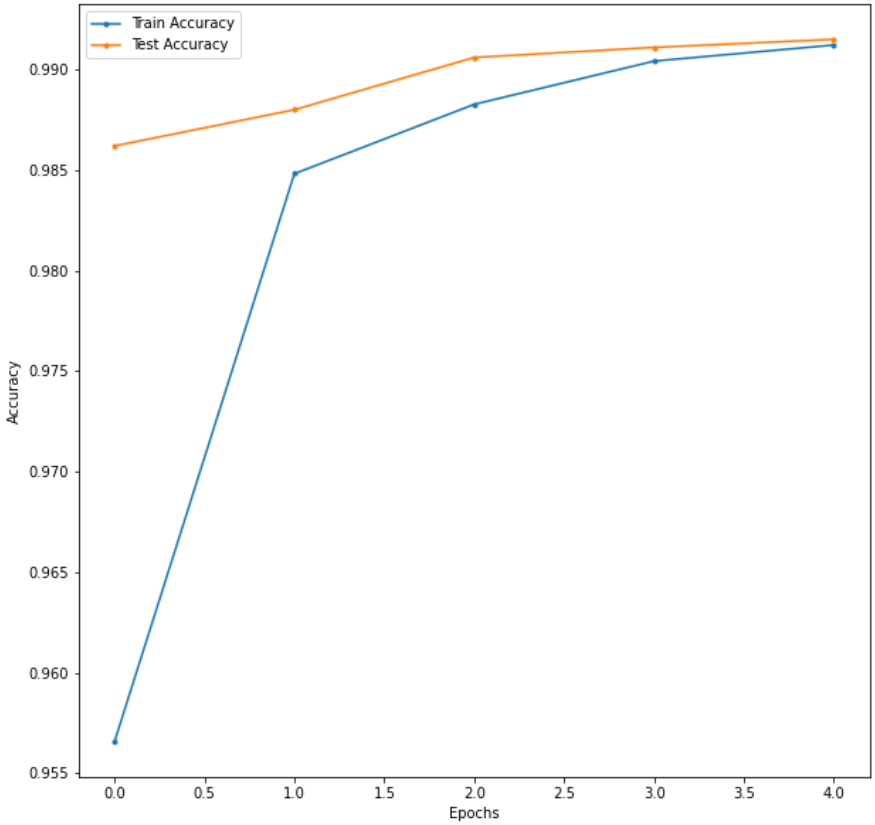


Fig (8): Digit Recognition Model Accuracy

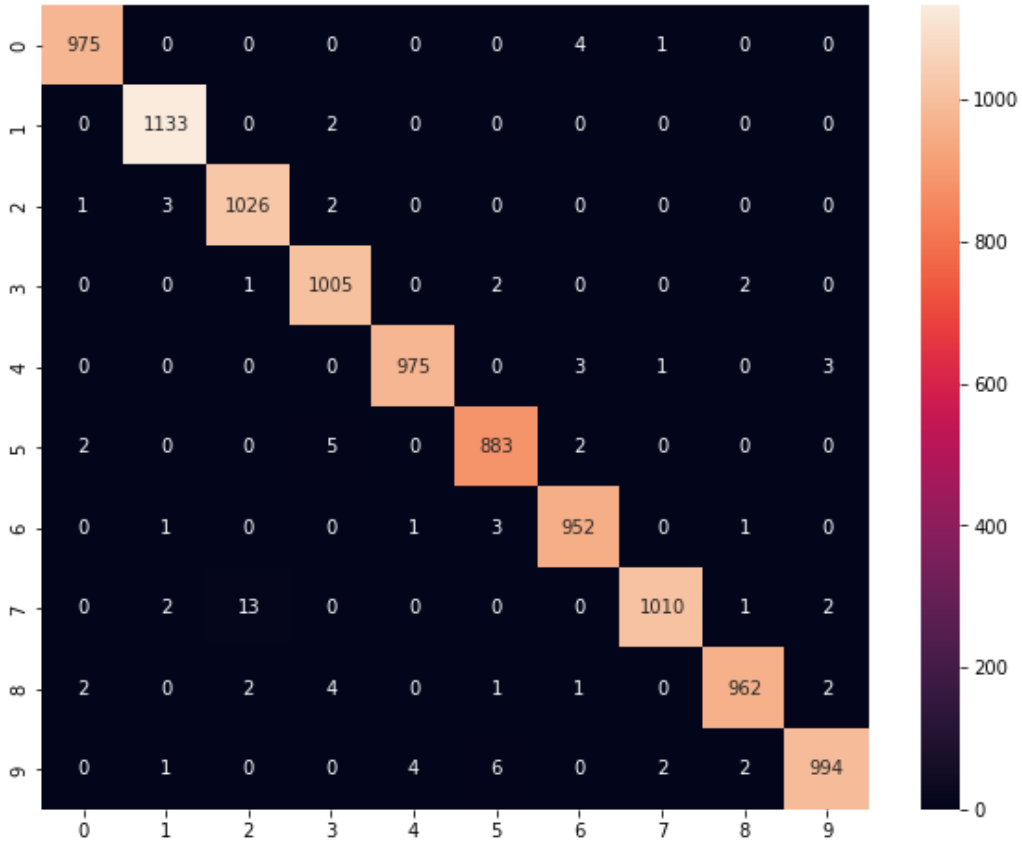


Fig (9): Digit Recognition Model Confusion Matrix

When we send each box to the ML model. We get the result of the grid of numbers, and zeros for empty boxes.
A link to the colab document for training the model can be found at the references section.

4.2 Algorithms

The robot operates in two modes: solve and puzzle.

4.2.1 Solving

For the solving algorithm, we used a backtracking algorithm, which works by looping through the grid and for each empty cell, try the numbers 1-10 in order, if a number fits all conditions of sudoku (no repetition in row, no repetition in column, no repetition in sub-box) go to the next empty cell and try for that. When a number fails, the algorithm tries the next number in that box. If we reach the number 9 and still fail, we backtrack to the previous algorithm-filled box and try the next number. Theoretically, this backtracking can be done for a max of 9 times for each row until it is filled.
To get the solution that we need to print, we subtract the original grid from the filled grid, which yields a list of the numbers to write, and zeros in place of already filled boxes. This solution is what we print.

4.2.2 Puzzle generating

To generate a puzzle, we first need to know the difficulty level, as the level of difficulty gets harder, the numbers in the puzzle grid are less. The ranges selected in our case are (30, 40), (18, 25), and (8,15) for levels beginner, intermediate, and advanced respectively. The algorithm fills random blocks with random numbers between 1 and 9 ensuring that with each number, the conditions of the puzzles are satisfied.

4.3 Mechanical Part

This subsection explores the mechanical details and parts that work together to deliver the output to the user. This includes the CNC printing machine.

For the microcontroller, we used an Arduino Uno with GRBL library modified to match the needs of our CNC machine.

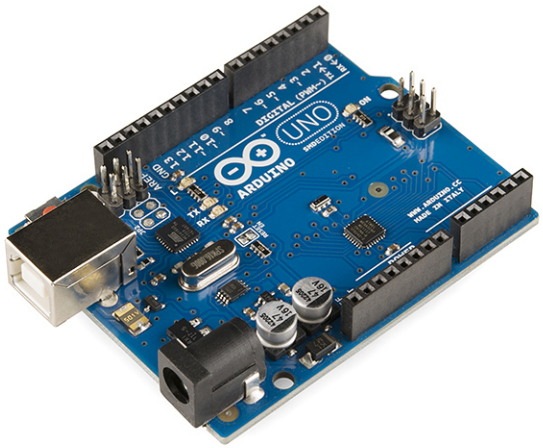


Fig (10): Arduino Uno

For the CNC machine itself, we used motor drivers of type A4988. This driver has an output drive capacity of up to 35V and ±2A. Which let us control a bipolar stepper motor such as the NEMA 17 at up to 2A output current per coil. The driver has a built-in translator for easy operation. For the motors, we used two stepper motors of type (STP-42D2131) and a servo motor. The stepper motors move the machine in the x and y direction and the servo moves the pen in the z direction as shown in Fig ().

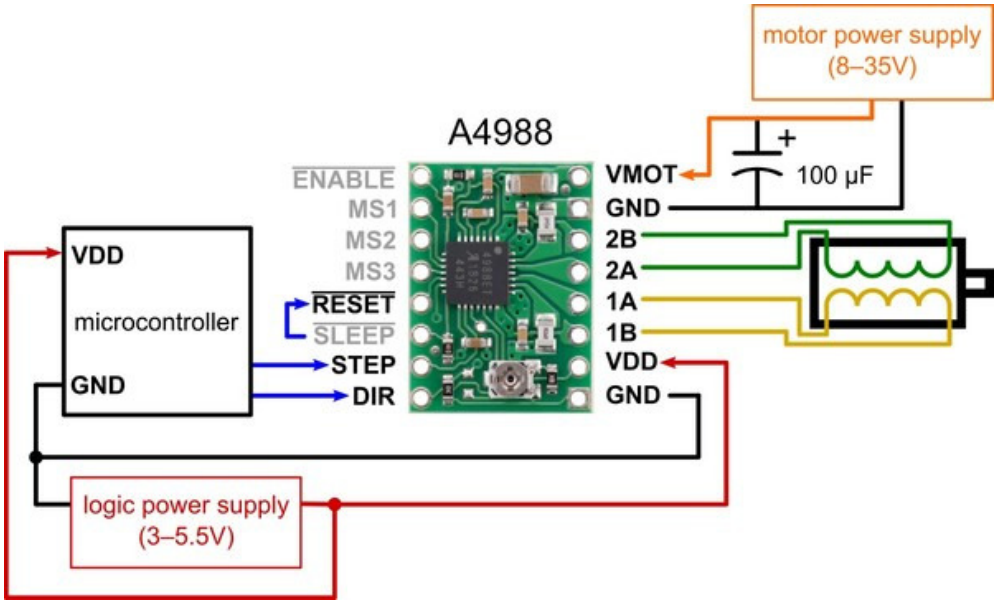


Fig (11): A4988 Driver Connection

Fig (12): STP-42D2131 Stepper Motor

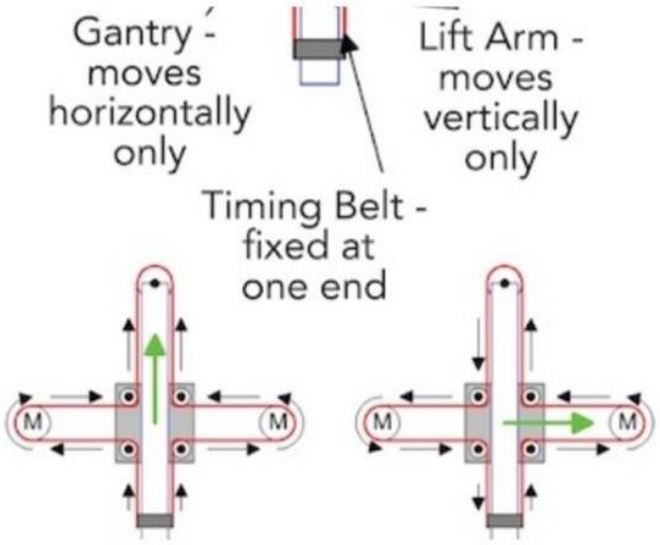
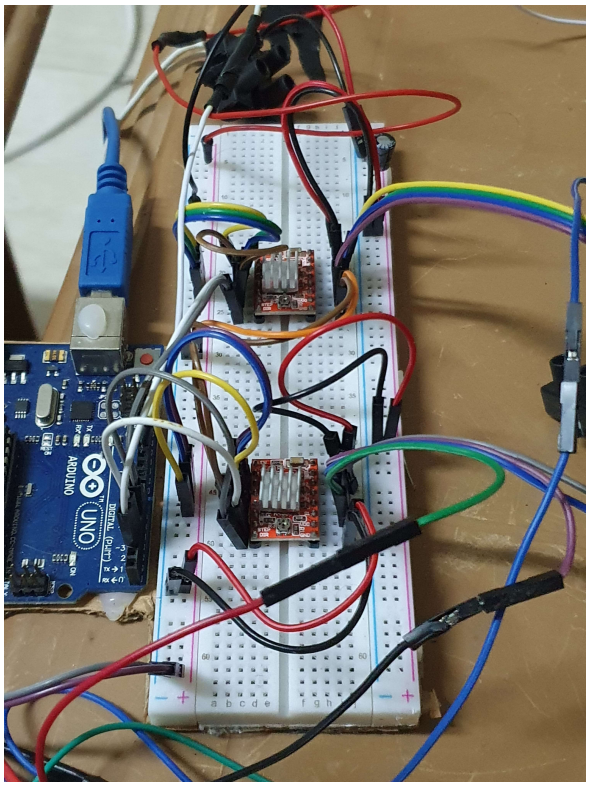


Fig (13): Movement of CNC in X and Y direction

Fig (14): Servo Motor

Other tools include wires, capacitor, and breadboard



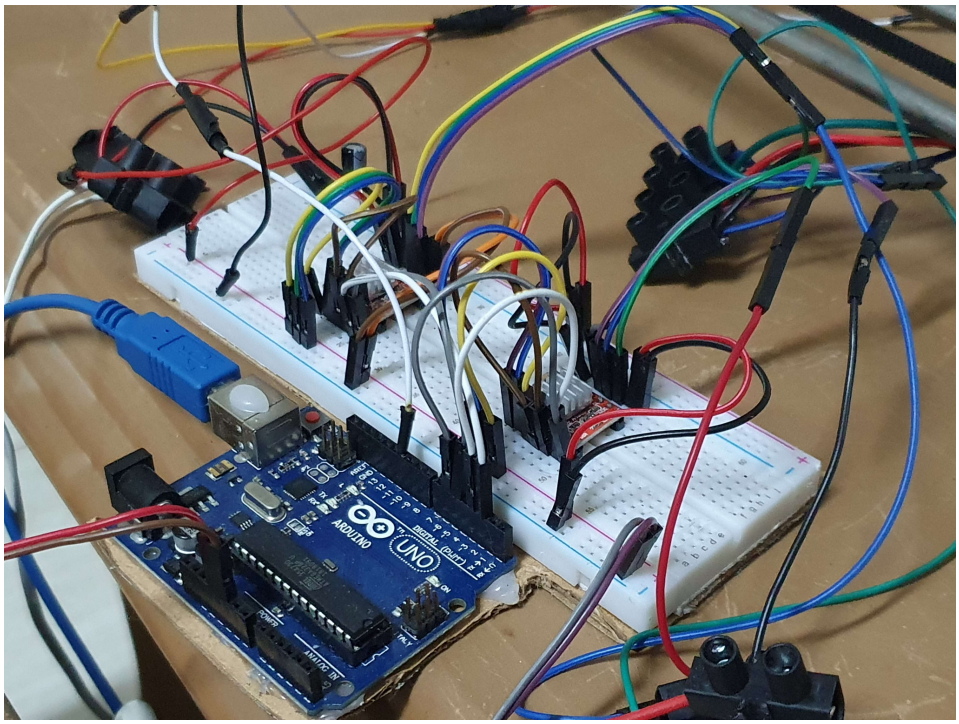
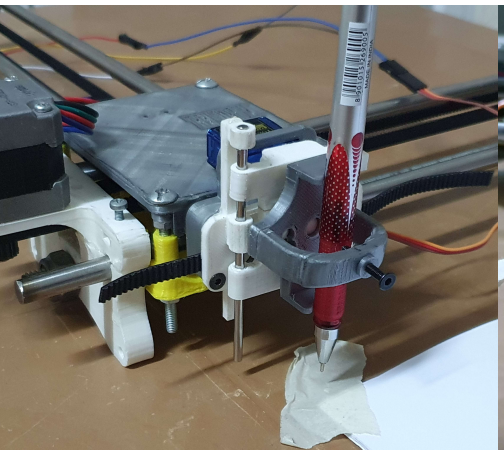
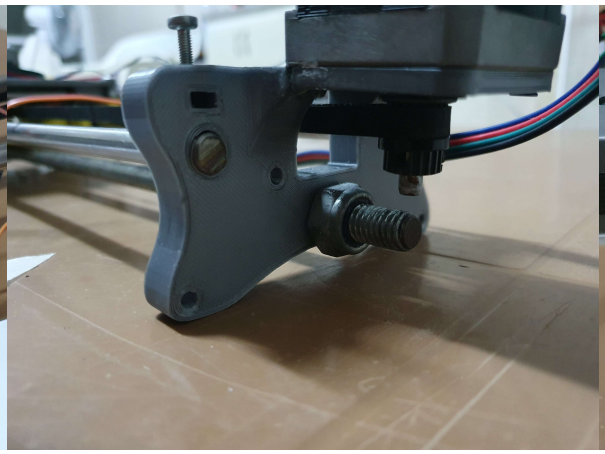


Fig (15): CNC Controller Connection

There are other 3D printed parts for the machine.





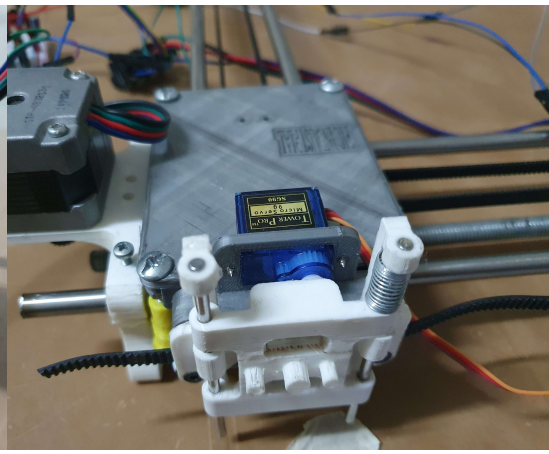


Fig (16): 3D Printed Parts

Other parts shown below include: Bearings, 8mm steel bar, belt, 3mm screw, 3mm nut, 4mm screw, 4mm nut, threaded bar, 10mm nut

Fig (17): Other CNC Parts

We used the Inkscape application and the Gcodetools extension to generate Gcodes for the numbers one through nine. We also used the Universal GCode Sender application to get Gcodes for the movement of the machine in different directions. Then, these Gcodes were stored on the memory card of the Raspberry Pi to be accessed with a script to move the machine according to the Sudoku matrix.

4.4 Mobile Application

4.2.1 Introduction

We wanted a simple way to control and send commands to the CNC machine through the Raspberry Pi. We created a simple android application which is designed to send string messages to Raspberry pi and receive messages from it. This application is responsible for controlling the CNC machine through the Raspberry Pi. The application contains three pages, one main page and two secondary pages for modes of operation which are solve mode and puzzle mode. We are going to describe every page and element used in this application. We used Android Studio to create this application because we have android devices and it is the easiest available method to create an android application.

4.2.2 Used languages and their usage

1. XML: It is mainly used for designing user interfaces, it has android buttons which are used to start programmed events. There are multiple edittexts which are used to enter sudoku input simulating the original sudoku board.

2. CSS: A commonly used language in frontend design used to customize and edit the style of xml elements to make them look better.

3. Kotlin: The main language used to program intended actions we need to make our application function well. It is the most common android development language currently replacing java language.

4.2.3 External libraries used

RetroFit library which turns REST APIs into java/kotlin interfaces so we can initialize REST API requests between Raspberry Pi and Android phone using ESP32 controller. The ESP32 acts as a connection line between the Android device and the Raspberry Pi.

4.2.4 Application description

1.Main Page:-

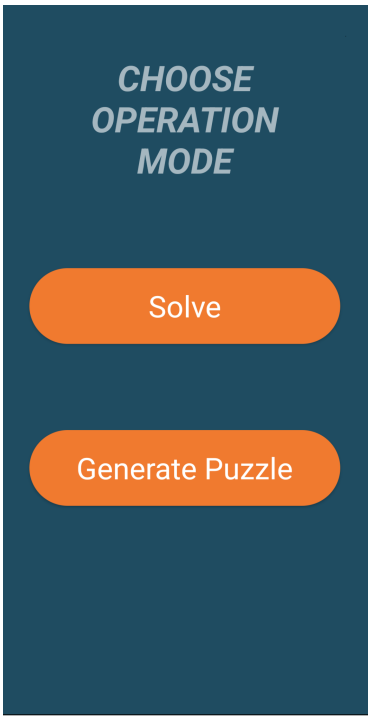


Fig (18): Mobile Application Main Page

This page has two buttons which will make the user choose between two options of modes which are solve mode and generate puzzle mode. The title “CHOOSE OPERATION MODE” is a textview which is used mainly to show any string and it is customizable using css styles and xml attributes. When the user presses the Solve button its on click listener will run which will change the current page to solve mode page. This occurs by changing the current activity from MainActivity to SolveActivity which will mainly be used to send commands to solve sudoku boards.

2.Solve Page:-

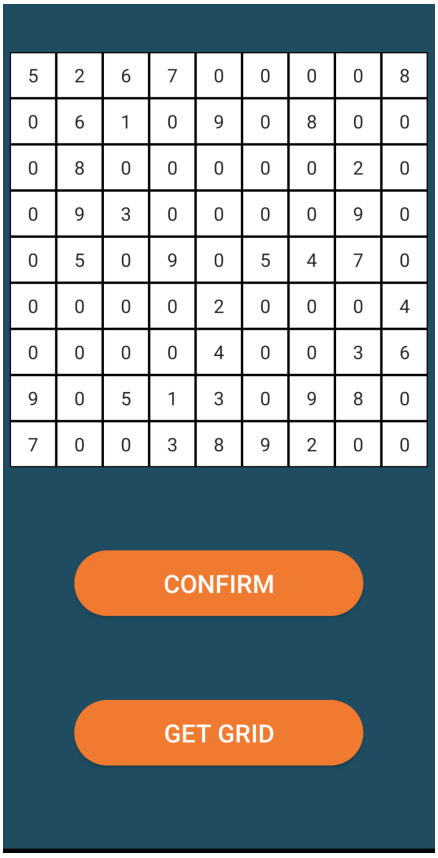


Fig (19): Mobile Application Solve Page

This page contains a 9X9 matrix of edittext which acts as a solve page main property. These edittexts can be modified, there are a total of 81 cells as in the 9X9 size of sudoku game. When the user presses the GET GRID button, the matrix will be filled with the received string from the ESP32. When the Confirm button is activated an event listener will handle the clicking action which will send a message to the ESP32 to initialize the printing process on the CNC machine.

3.Generate Puzzle Page:-

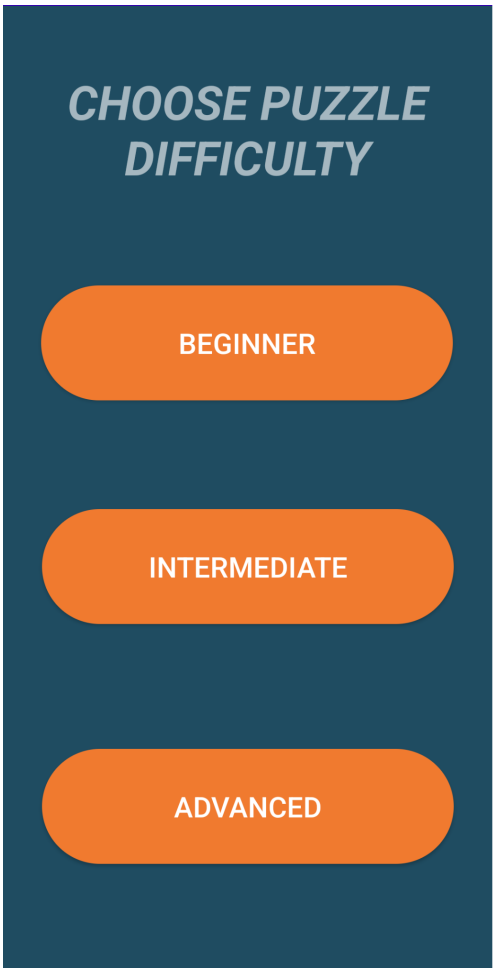


Fig (20): Mobile Application Generate Puzzle Page

This page contains three buttons which are used to set the difficulty of the generated puzzle. Whenever a button is pressed the application will send a corresponding message to ESP32 according to the selected button. There is an onClick listener for each button which will handle the pressing action which will send a string to the ESP32.

4.2.5 How does the application work

This application relies on REST APIs to send and receive data from and to the ESP32 to control the CNC machine. There are predefined key values on the ESP32. The application will use these values to send data to the ESP32 to make it recognize what the transferred data is. There are three defined values to differentiate every mode: mode, values and level.

- mode: there are three values for mode which are: take image, solve and Generate Puzzle.
- values: used to fill the grid in solve mode and send the grid values to ESP
- level: used to select the difficulty in Generate Puzzle mode.

When the user starts the application the main page will be shown. If the user chooses the Solve button they will be taken into the solve mode page the camera will capture the grid, then the user has to manually edit wrong numbers on the grid so the puzzle will be solvable. After confirming the input the application will specify that the mode is Solve Mode and values are as in the matrix and send the data to the ESP and from there to the Raspberry Pi to initialize printing operation.

If the user chooses Generate Puzzle mode, the Generate Puzzle mode page will open when the user chooses the difficulty, the application will send data to ESP with the selected difficulty and to the Raspberry Pi to initialize the puzzle generating process and print it.

4.5 Communication

In this project we needed to be able to communicate between the Raspberry Pi and the CNC machine, for that we used an Arduino Uno. To communicate between the Raspberry Pi and the mobile application, we used an ESP32 module with built in WiFi. The mobile communicates with the ESP32 and the ESP32 communicates with the Raspberry Pi. This application contains three pages: one main page and two secondary pages for modes of operation.

Fig (21): ESP32

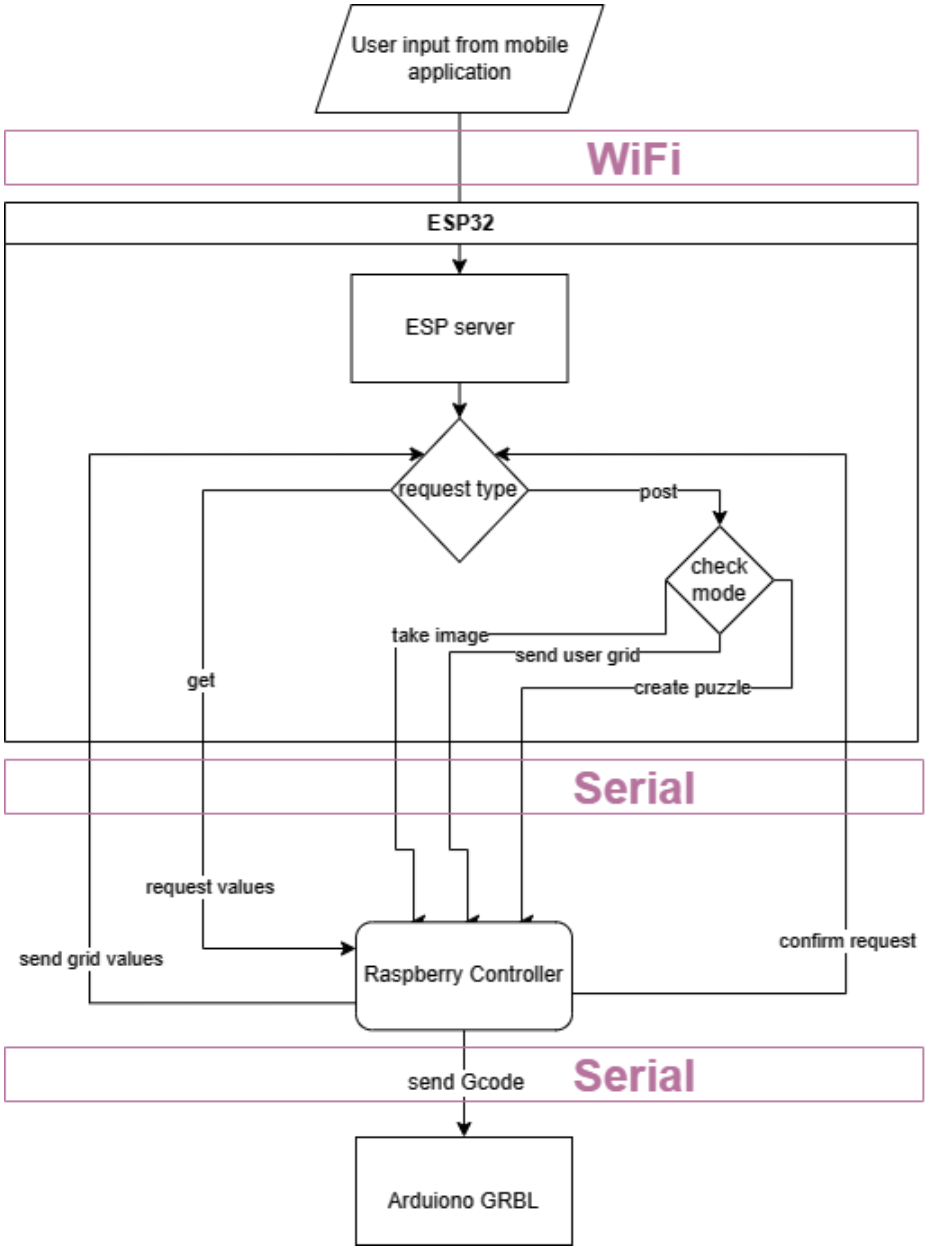


Fig (22): Overview of the Communication Process

Chapter 5. Results and Discussions

5.1 Testing

We tested the machine by using handwritten numbers on the printed Sudoku board on the paper. We tried to write numbers in a way to make them as easy as possible to be recognized by the image processing module. Although the module isn't very accurate so there were many errors while getting numbers from the image.

Sometimes the motors get stuck while working so we have to manually adjust them. We used a software called Universal Gcode sender and installed the GRBL library on the Arduino. We used the software to send commands to the GRBL library to control the motors. To test the motors we connected them with the Arduino to make sure they are working as intended. We tried to adjust the distance between the pen and the paper so the writing process was as accurate as possible. We faced many problems during the testing process especially on the ESP32 device but we managed to solve them.

We decided to create a small application to help in testing and in the final project. This application is now the main method of interaction between the user and the machine.

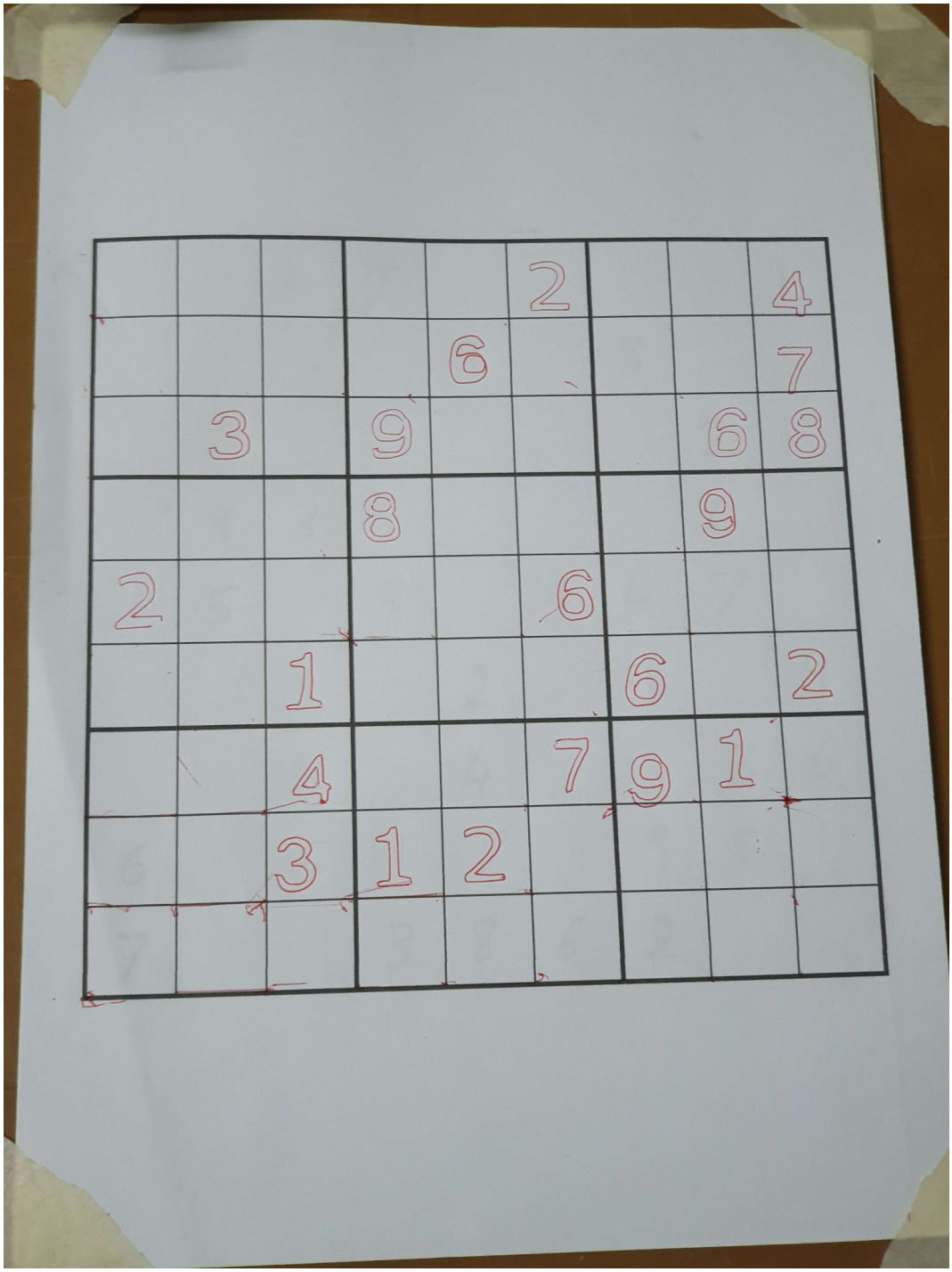


Fig (23): Result of the Machine in Puzzle Mode

Conclusion and Recommendations

Finally, after going through the troubleshooting stage of the project, we managed to get the machine to work. It is not the best result, but it is sufficient enough for a low-cost small scale project. We tried to get the error rate to be as low as possible, whether it is in the CNC machine motors or in the communication between parts.

Since the final result of the project is not perfect, further optimizations and improvements can be done, here are some recommendation that we would like to suggest:

1. Since the motors were giving us issues with accuracy, we recommend the use of better and more accurate motors. This will reduce the error rates and will help make the machine faster. This was expensive for us, but in case of a bigger budget, this is definitely the better approach.
2. Getting a newer version of the PiCamera. The version we managed to get is an older version and there are newer versions with better colors and better image resolutions, which could make the machine learning model perform better.
3. Making a better machine learning model for digit recognition. Although the model we made had a high accuracy when trained, it proved to be less accurate in practice, this could be because of the variance of handwriting and this is a common issue with deterministic models.

Here is the future work we are planning to do:

1. Adding a verify mode. Which can read an image of a full grid and tell the user if the solution is correct and passes all puzzle conditions.
2. Adding a hint mode. Which lets the user choose a certain cell in the grid to get the right number in that cell and print only that box.
3. Further improvements to the mobile application. These include a better UI/UX for the user and more handling for some cases.
4. Adding a display to the machine so the user can see the state of operation.
5. Adding other control methods other than the mobile application so the user can use the machine directly.

References

- *ArduinoJson: Efficient JSON serialization for embedded C++*, https://arduinojson.org/
- *Kotlin Programming Language*, https://kotlinlang.org/
- *OpenCV*, https://opencv.org/
- “Arduino Software.” *Arduino*, https://www.arduino.cc/en/software
- “AsyncTCP.” *GitHub*, https://github.com/me-no-dev/AsyncTCP
- “Draw.io to create diagrams.” *diagrams.net*, https://app.diagrams.net/
- “ESPAsyncWebServer.” *GitHub*, https://github.com/me-no-dev/ESPAsyncWebServer
- “GRBL with Servo.” *GitHub*, https://github.com/bdring/Grbl_Pen_Servo
- “ML model training notebook.” https://colab.research.google.com/drive/1D6e2K0Dsbwe7Dp3r9jD4Qo_dzsKqsIN_?usp=sharing
- Pandian, Sundar. “Journal of Scientific Research & Engineering Trends.” *Journal of Scientific Research & Engineering Trends*, https://ijsret.com/wp-content/uploads/2019/05/IJSRET_V5_issue3_342.pdf
- “Universal Gcode Sender.” https://winder.github.io/ugs_website
- N. Karnani, P. M. Tiwari and J. K. Rai, "*Design and implementation of sudoku solving robot using microcontroller*," 2013 Students Conference on Engineering and Systems (SCES), Allahabad, India, 2013, pp. 1-5, doi: 10.1109/SCES.2013.6547572.