**CHAPTER 1**

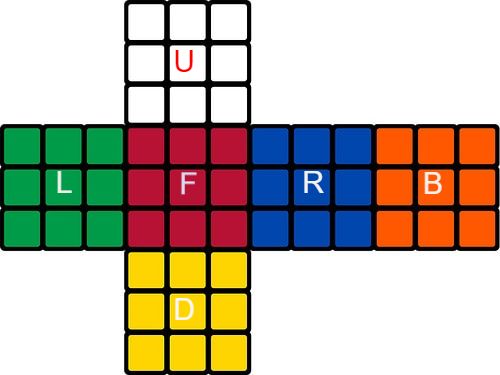
# INTRODUCTION

**1.1 Project Overview**

**Rubik's Cube Solver (RCS)** is a complete program that can solve any scrambled 3X3X3 cube in less than 22 moves. It uses kociemba algorithm for finding the most optimum solution of a scrambled cube.

RCS use**OpenCV** for image acquisition and Image Processing. Image can be acquired from a webcam. All the faces of the cube are scanned successively. The image is stored using **HSV** color space. Color of each face is detected as soon as the face is properly aligned. Further the color data is processed to find the position of different colors on the cube. This data is passed on to the Kociemba Rubik's Cube Solver Algorithm.

The Kociemba Algorithm generates a table of moves during the first run on a system. On further sessions the algorithm goes through all the possible moves that can solve a cube. It finds the moves which will solve the cube in shortest possible time. The program ensures that the cube is solved in less than 22 steps. These moves are simulated using the GUI.



**Fig 1.1 Faces of Rubik`s cube**

**1.2 Purpose of Project**

The most vital a part of solving a Rubik's Cube is knowing how it works Rubik's cube is one in every of the foremost challenging and most famous puzzles of all time.

It continues to stay within the top among the puzzle games thanks to the look of the puzzle i.e., one correct solution out of the 43 quintillion other possibilities.

Only 5.8% of world population can solve the Rubik's cube efficiently. But Al& ML concept is ready to try and do with minimum number of moves.

There are three major parts to solve any Rubik's Cube First is identifying the positions of various colors at different positions.

Second is to develop a series of steps which may be wont to solve the cube and third is to implement these steps on the cube to induce the ultimate result.

**1.3 Scope of Project**

Our end goal is to build a system that is able to read the state of the Rubik’s Cube reliably, find a solution and then solve the cube using a robot. We want to demonstrate the challenges associated with building puzzle solving robots. From the real world challenges in vision and robotics down to the theoretical challenges that lie in the search space, this is not a trivial task.

**1.4 Proposed System**

In order to beat the prevailing problems, we propose the way to solve “Rubiks Cube Using Open CV”. Aim of this project is to solve a unsolved Rubik's Cube. The program contains OpenCV libraries and that we are solving the cube using Kociemba algorithm.

We decided to use Kociemba’s algorithm rather than the optimal solution because it gives us a “reasonable” solution during a short amount of computing time. Kociemba’s algorithm guarantees an answer with under 30 moves.

Kociemba’s algorithm creates a two-phase approach that permits it to go looking for an answer faster the primary phase is to induce the cube to “state 1.” State 1 is any subset of the cube which will be achieved by running the moves: Up clockwise, down clockwise, Right 180, Left 180, Front 180 and Back 180 from a solved cube.

Kociemba’s algorithm uses an intensive set of pruning tables to formulate the foremost efficient “move set” to rework a scrambled cube to state 1.

Then, it uses the identical moves given above to remodel the cube from state 1 to a “solved state.”

When a unsolved Rubik's cube is shown before of a camera, and therefore the computer vision algorithm detects the position of the face within the webcam picture and identifies the facelet stickers along with their colors.

The two-phase-algorithm is used to solve the Rubik’s Cube.

When these moves are followed we get the tip product of a solved Rubik's Cube within 30 moves.

**1.5 Existing System**

Rubik’s Cube could be a widely popular mechanical puzzle that has attracted attention round the world due to its unique characteristics. As a classic brain-training toy well-known to the general public, Rubik’s Cube was used for research project and technology development by many scholars.

The Rubik’s cube could be a single-player game and a sequence of random moves, irrespective of how long, is unlikely to finish within the goal state. Developing machine learning algorithms to cater to this property of the Rubik’s cube might provide insights into learning to unravel planning problems with large state spaces. Although machine learning methods have previously been applied to the Rubik’s cube, these methods have either did not reliably solve the cube1–4 or have had to depend upon specific domain knowledge.

Outside of machine learning methods, methods supported pattern databases are effective at solving puzzles like the Rubik’s cube, the 15 puzzle and also the 24 puzzle7,8 but these methods are often memory-intensive and puzzle-specific. More broadly, a significant goal in AI is to make algorithms that can learn the way to master.

The classical 3×3×3 Rubik’s cube is barely one representative of a bigger family of possible environments that broadly share the characteristics described above. because the size and dimensions are increased, the complexity of the underlying combinatorial problems rapidly increases. for instance, while finding an optimal solution to the 15 puzzle takes but a second on a modern-day desktop, finding an optimal solution to the 24 puzzles can take days, and finding an optimal solution to the 35 puzzle is mostly intractable.

**Summary**

This section mainly deals with the introduction of Rubik’s cube project, it defines the purpose of project also it defines the system we are going to develop in our project.

## CHAPTER 2

**LITERATURE SURVEY**

* 1. **Advanced Rubik’s Cube Algorithm Solver**

#### **Authors: Vasile** Dan, Gabriel Harja, IoanNașcu

**2.1.1 Hardware Implementation**

The process of solving the Rubik’s cube begins with the acquisition of data from 4 webcams. They are positioned in four different corners of the cube at 200 mm. Two of the webcams have been positioned at the top and the other two at the bottom, thus all the pieces are visible.

The colors on each side of the cube are identified and the two implemented methods, Kociemba’s algorithm, and blindfolded method, are applied. The generated solutions are sent to Arduino Due via the serial interface. Six stepper motors will rotate the faces of the cube, according to the chosen method. The 4 digits of the 7-segments display will show the solving time.

**2.2.2 Color Recognition:**

* Converting the color space of the image: from BGR (Blue, Green, Red) to HSV (Hue, Saturation, Value). The other two parameters are used for variating shadow andshine.
* Image filtering: using openCV library, Smooth Median filter, for reducing the salt and pepper noise.
* Selectingtheregionsofinterestfromthefourimages.Theseregionsconsistofonepixelon piece of the cube, due to the previously appliedfilter.
* Color identification, based on the 3 values of thepixel.

**2.2 Autonomous Rubik's Cube Solver Using Image Processing**

**Authors:**Harshad Sawhney, SakshiSinha, AnuragLohia, PrashantJalan, PriyankaHarlalka.

Mechanical design: One of the most important parts of the robot was its mechanical design. On each of the four side walls of the cuboid, One stepper motor is mounted with utmost precision so that motors at opposite faces has their shafts coaxial. On each of these motors a gripper has been mounted which is driven by servo motors.

Electronics: The electronic part basically contains a microcontroller which is the brain of the robot. The microcontroller used is “Arduino Mega” and the motors are connected to the pins of the microcontroller which gives commands to motors based on a code stored inside the chip or at mega of the microcontroller.

Image processing: Image processing is a technique to store the colours of any object in a format that can be processed through codes and programs

Software Architecture: Image is being processed by the code of Dev C++. We are taking the images of all the faces of the cube. The sequence in which the image is taken is determined by the kociemba algorithm.

Binary Conversion of Image: A 3-channel RGB image is captured by the camera. Each color has its own three values (hue, saturation and value) which distinguish it from other colors. Using this property, a binary image can be generated for all the six colors of the cube. Centroid Detection & Filtration: Blob detection is applied on the binary image of each color. CV Blob library is provided by OpenCV which is used for detecting the blobs.

Koceimba’s algorithm: After finding the matrix that stores the colors of the individual blocks of the six faces of the cube, an algorithm are required which provides the steps that can solve the cube. Arduino coding. The Arduino is coded in Arduino interface and it provides rotation to each face of the cube. The code ensures that the grippers do not collide with each other at any point of time while unscrambling the cube

## 2.3 Rubik’s Cube Solver: A Review

## Authors: Ms. Ekta S. Toshniwal, Mr. Yogesh Golhar

**Algorithms used:**

Thistlethwaite’s algorithm: The Thistlewaite Algorithm was the least effective method of solving

the Rubik’s cube requiring an average of 42 moves and 3 mins and 48 seconds. This method was

found to be not too much effective and useful.

* Kociemba’s Algorithm: Kociemba’s Algorithm was an improvement on Thistlethwaite’s algorithm. The Kociemba’s algorithm proved to be the 2nd most effective algorithm, requiring an average of 28 moves and 2 mins and 32 seconds.
* Korf’s Algorithm: The Korf’s algorithm proved to be the most effective algorithm, requiring an average of 20 moves and 2 mins and 5 seconds. Korf’s algorithm is based on a multi-phase coding, which means the equations are divided into numerous sub problems which are then solved.

**Programming:**

The programming consists of image processing concept, somewhat pattern recognition, various algorithms for finding the solution of the scrambled Rubik’s cube and coding of microcontroller to direct the motors accordingly. The microcontroller used in this project is Raspberry Pi. Programming language used is Python. Image Processing is used in order to recognize the colors on each faces of the cube, image processing concept is used.

**2.4 Overview of Rubik’s Cube and Reflections on Its Application in Mechanism**

#### **Authors**: DaXingZeng, MingLi, Juan‑JuanWang, YuLeiHou, WenJuanLuand Zhen Huang

This base paper helped us to understand and analyze the drawbacks commonly faced in solving the Rubik's cube.

Every method used to solve the Rubik's cube involves machines like stepper motors, grippers, rotators, and other electronics. There isn't any model designed yet to solve it manually without the use of machines. We have used the concept to build such

Software which could solve the Rubik's without the use of any hardware like machines we can use it help the user solve it with his own hands. This reduces the cost of construction of the design and makes it more interactive and user can analyze the steps and the logic involved in solving such puzzle.

**Summary**

This section lists all the literature surveys used in the project. It explains all the surveys with all the key points and the algorithms used in it.

**CHAPTER 3**

**SYSTEM DESIGN**

Design is an innovative procedure; a successful system is developed through great designs. System design is a process of giving detailed information about the proposed work in a physical format. Different designs are built for development of system, which describes about features, components which are included and how client interact with system.

**3.1 Fundamental Design Concepts**

Fundamental design is developed in course of recent years. As year’s passes, enthusiasm of creating new designs is evolved and each design has been tested. Software designer gets new ideas and foundation to build and test new design concepts. Fundamental framework is design to "getting it right". Major plan ideas, for example, deliberation, refinement, modularity, programming engineering and data encryption is applied to meet the requirement of proposed work.

**3.1.1 Input Design**

Input Design is a way toward changing user-based inputs into computerized format. Main objective of input design is, to make computerization as possible and error free. Giving a decent information configuration to the application simple information and determination highlights are received. The input design prerequisites, for example, ease of use, reliable organization, and intelligent exchange to help client to get proper information on time. Input design is a general framework which exceptionally cautious consideration. Gathering all input parts is one of the costly parts of framework.

**3.1.2 Output Design**

Output design meets the necessities of client and presents the output data clearly. In any framework processing result are conveyed to clients and different frameworks in form of output design. It is direct source to client. Productive output enhances framework association with machines of source and destination. Output is in format of a Pie Chart which is get for

the proposed system. The optimized Pie-Chart include the percentage of the individual genre shadowing the news article.

**3.2 Development of System**

Development of system is a method, where development of product is completed, or it solves all problems of system. Development of software includes number of stages and process to develop software. Step by step procedure is followed to complete development of software. The method which is followed in this project is incremental model.

**3.2.1 Different phases of model**

* **Requirement:** This phase includes collection of all requirements which is needed for development of software.
* **Design:** The specification of system is converted into software design, by keeping in mind system specification. The designer mainly describes about algorithm, architecture, and structure of system.
* **Coding:** Developer begins coding with a specific end goal to give a full outline of project. As such framework system specification are just changed over into system decipherable process code
* **Implementation:** This stage includes execution of project which involves coding. The output is commonly a library, documentation, and client manuals.
* **Testing:** The testing stage includes all modules of project which is coordinated and tried to guarantee that total framework meets product necessities. Verification and Validation is mostly concerned under this testing phase.
* **Maintenance:** It is most important stage, where product should be user friendly, adaptable, error free and should improve productivity of project.

**3.2.2 Purpose of choosing Increment model**

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.The system is put into production when the first increment is delivered.

The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments.



**Fig 3.1 Incremental Model**

**Characteristics of an Incremental module includes**

* System development is broken down into many mini development projects
* Partial systems are successively built to produce a final total system
* Highest priority requirement is tackled first
* Once the requirement is developed, requirement for that increment is frozen

**When to use Incremental models?**

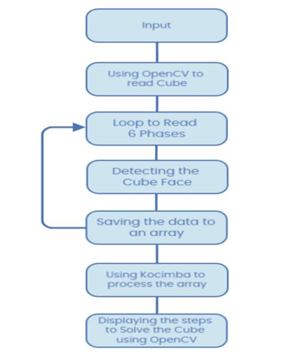
* Requirements of the system are clearly understood
* When demand for an early release of a product arises
* When software engineering team are not very well skilled or trained
* When high-risk features and goals are involved.
* Such methodology is more in use for web application and product-based companies.

**3.3 System Architecture**

System architecture describes about each component of system.

The modules in this project are shown below:

* Open CV: OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Centroid Detection & Filtration: Blob detection is applied on the binary image of each color. CV Blob library is provided by OpenCV which is used for detecting the blobs.
* Koceimba’s algorithm: After finding the matrix that stores the colors of the individual blocks of the six faces of the cube, an algorithm are required which provides the steps that can solve the cube.
* Arduino coding: The Arduino is coded in Arduino interface and it provides rotation to each face of the cube. The code ensures that the grippers do not collide with each other at any point of time while unscrambling the cube.

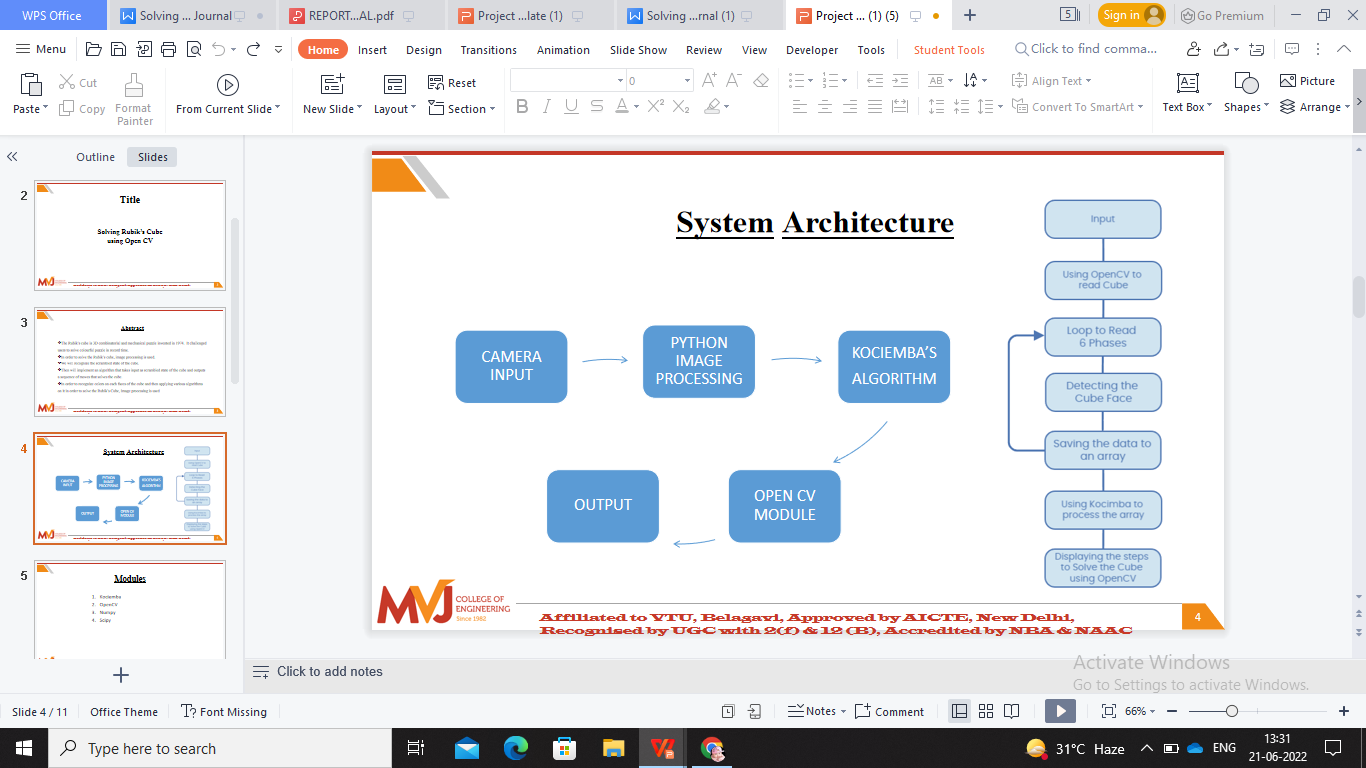
****

**Figure 3.2 System Architecture Diagram**

**3.4 Sequence Diagram**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams

are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.



**Figure 3.3 Sequence Diagram**

**Summary**

This chapter mainly concentrates on input design, output design, waterfall model, system architecture and various designs which is involved in proposed system.

**CHAPTER 4**

**IMPLEMENTATION**

It is main part of project where project is implemented. Implementation must be clearly defined, planned carefully and systematically otherwise it causes confusion and leads to generation of problems. Following tasks involved in stage of implementation:

* Planning carefully
* System Investigation.
* Investigating different types of systems and constraints.
* Selecting the most accurate and correct language for developing the application.
* Understanding and correctly evaluating several changeover methods.
* Rightful decision is made about the selection of the platform.

**4.1 Language used for implementation**

Implementation is an important step where execution of project is done. Hence language used for implementation should be adaptable, error free, user friendly etc. Most of the time because of some mistake projects get ruined or spoiled due to inappropriate programming language.

Python is widely considered as the preferred language for ML (Machine Learning). Few simple reasons are:

* **It has an excellent collection of in-built libraries**: Python claims a huge number of in-built libraries for data mining, data manipulation, and machine learning. For example, NumPy, this is used for scientific calculation. Scikit-learn have tools for data mining and analysis which optimizes the Python’s brilliant machine learning usability. Panda is another package which provides developers with high-performance structures and data analysis tools and helps them to reduce the project implementation time. Similarly, SciPy is used for advanced computation, and Pybrain is used for machine learning.
* **Moderate learning Curve:** Python Programming language is very accessible and easy to learn and use. It focuses on code readability. It is versatile and well-structured language.
* **It is a general-purpose programming language:**Well, Python is a good choice if the project requirements are more than just information such as developing afunctional website.
* **Easy to integrate:** Python programming language incorporates better than “R” in business environments. It is easy to integrate Python with other lower-level languages such as C, C++, or Java. Similarly, the Python based stack is easy to incorporate with data scientist’s work, which allows it to bring efficiently into production.
* **High productivity:** Syntax in Python is extremely readable and easy to understand other programming languages, whereas R programming language has different syntax. The readability syntax in Python programming language confirms high productivity of development teams.
* **Easy to create prototypes:** It is already stated that Python is easy to learn and fast to develop. It requires less coding, which means that you can create prototypes and test your concepts quickly and easily in Python as compared to other programming languages. Developing prototypes in Python not only saves developers time, but it also decreases your company’s costs.
* **It is free and open source**: Python programming language is available freely; this allows you to download Python for free, i.e., you can download its source code, make the modifications in it and then distribute it.
* **Object-oriented paradigm:** Python programming language supports for both the Object-oriented and procedural programming models. Classes and objects in Object-oriented programming help us to model the real world while functions in procedural programming enable us to reuse the code. A class in Object-oriented Programming encapsulates data and function together.
* **Portability:** Python programming language is portable, i.e., code written in Python can be run on another platform; this is called Write Once Run Anywhere, i.e., WORA, but this is not possible with other languages like C++.

**4.2 Implementation Platform**

Platform plays an important role for development of software. A platform is a place where software is launched. Ubuntu platform is used for implementation of project. There are various analyses for choosing Ubuntu platform. It is an open-source operating system. It is used for many different purposes such as scrutiny, embedded system, security etc. it also contains remote connection and option to restore, so that file can be restored if deleted. Ubuntu software is used for scalable processing, multi-tasking, Encrypted File System (EFS) and smart card support. Google colab is used as the platform to perform the implementation.

**4.3 Color Detection**

# The dataset for our application would be each faces of the Rubik’s Cube. Also, each color in the cube is identified with unique number. So, when we show each face of cube to the camera it asks us to show each face of the cube. Like Front, Top, Left, Right, Bottom. So, it input as array of nine numbers in it with each number showing different color. We use Kociemba’s algorithm, and, we make use of Open CV and Image processing techniques in this project.

We check the regions that were created from the thresholding, discarding the regions that are not an acceptable size, do not have an acceptable shape (eccentricity), or are not solid. At this point, we should be left with only faces from the cube. However, some of the faces on the edge may have the black border visible and thus were not detected. We will need

to recover these faces by estimating their location from the detected faces.

# 

**Figure 4.1 Detecting Colors**

**4.4 Algorithm**

Kociemba’s algorithm performs much more better than Thistlethwaite’s as it reduces the quantity of phase transitions to only 2 rather than 4 this implies that transition must be applied between 3 groups:

G0 = hU, D, R, L, F, Bi.

G1 = hU, D, R2, L2, F2, B2i

G2 = {C}

**Group G0**

G0 is that the group of all states reachable using moves L,R,F,B,U,D. Notice how this is often just all reachable states using any of the legal moves defined in section 2.1.3 since we are able to perform any L2,R2,F2, etc moves by changing L \* L , R \* R, F \* F. Similarly we are able to perform any L3, R3, F3, by changing moves L \* L \* L, R \* R \* R, F \* F \* F. Our aim is to maneuver from G0 → G1 → G2 → G3 → G4. Where G4 contains only the solved cube state.

**Group G1**

G1 is the group of these moves L,R,F,B,U2,D2 to reach all the states .G1 is different to group G0 as all smaller states are reachable. G1 contains only ‘good’ edges. to work out why this can be so, allow us to recall to our edge flip lemma in section.

2.2.1.2. to clarify why there are always an excellent number of flips, we proved that using only moves U, R, D and L, it's out of the question to flip any edges. rather than moves U, R, D and L, allow us to prove the identical result's possible using moves L, R, F, B, U2 and D2.

**Group G2**

The group G2 is simply the solved state. Using only moves in G1 the state is changed directly from G1 → G2.

**4.5 Hardware Implementation**

One of the most important parts of the Rubik’s cube solver is its mechanical design. To design a system with proper and accurate measurement is essential for proper

working of the system. The mechanical design requires a very high precision .Even the slightest error of few millimeters can result in failure.

When looking at a Rubik's Cube, there are six sides, each containing nine pieces. The sides can be rotated in many ways, the centre pieces don't move with respect to each other. Therefore, when the cube is being solved, the central pieces cannot move position. The initial state of the Rubik’s Cube is determined through a camera that captures each of the six faces of cube. Then an array of data is produced which contains location and colour of all the pieces, string is generated using that data. Then a solving algorithm is applied on it that takes as input the scrambled state of the Rubik’s Cube and outputs a sequence of moves which provides desired solution steps. As the algorithm for solving the Rubik’s Cube produced sequences of moves. Each move is then translated into a series of steps by one of the six stepper motors attached to one face of the Rubik’s cube. Each stepper motor which is controlled by stepper motor drivers turns a face of the cube. The stepper motors then execute the moves produced by the algorithm for solving the Rubik’s Cube completely. At the end, we will have a Rubik’s cube which is solved in minimum number of moves and in minimum time.

* **Stepper Motors**

To physically rotate and solve the cube we will use NEMA 17 bipolar stepper motors. Stepper motors are selected over other types of motors because they are continuous (i.e., they can rotate in one direction infinitely), they have a relatively high

torque, and they are relatively precise without additional feedback. The stepper motor operates using two coils and a permanent magnet rotor. When the coils are energized in a particular sequence, the stepper rotates by precisely 1.8degrees (i.e. it takes 200 steps for the shaft of the motor to make a complete rotation). There are six such stepper motors attached to six faces of the main frame i.e. all six faces of the scrambled Rubik’s cube which can provide the rotation of six faces at that particular instant and rotate in particular angle.

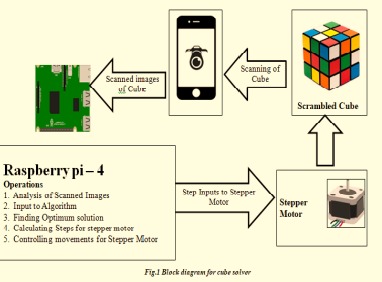
* **Motor drivers**

Rotating stepper motors is a nontrivial task and often requires a specialized driver. To reduce the control complexity, we use six stepper motor drivers to control six stepper motors. The stepper motors are connected through a L298 motor driver which acts as a switch and helps to rotate the motor. The motor driver pins are connected to the microcontroller which provides appropriate instructions to it to perform necessary actions. They have PWM (pulse width modulation) pins which can be used to control the speed of stepper motors and to instruct how much to move and in which direction to move.

* **Electronics**

The electronic part basically contains a microcontroller which is the brain of the robot or any mechanical design to solve the system. Without a microcontroller, it is not possible to solve it. The microcontroller used is “Raspberry Pi”. Stepper motors can be interfaced with any controller through the motor controller but the reason for using Raspberry Pi to control the stepper is its high computational power that makes it suitable for self-learning robots. The motors are connected to the pins of the microcontroller which gives commands to motors. These commands are based on a code/instruction stored inside the chip of microcontroller. The chip of microcontroller is very small but is used for storage purpose of instructions and plays a vital role. The motor driver pins are connected to the microcontroller which provides appropriate instructions to it. They have PWM (pulse width modulation) pins

which can be used to control the speed of stepper motors.Fig.1 shows block diagram of processing of Rubik’s cube solver

****

**Figure 4.2 Block diagram for cube solver**

**Summary**

The chapter describes about implementation process of proposed system. It describes about which platform, language, algorithms, and processing techniques used for executing the project.

**CHAPTER 5**

**TESTING**

Testing is an important part of project where testing of each module is done. Testing guarantees that proposed system is well organized analyzed to meet require project goal. Testing is last stage of project which guarantees the system is error free and ready to give desired output. The goals of testing are given by:

* Give operational quality to system
* Search and remove errors.
* Best quality project is produced.
* To approve the product as a solution for the first issue.

The following are types of testing performed in proposed system.

**5.1 Unit Testing**

Each module of project is tested individually. Verification is done on each module. Module of project is tested individually. Testing is done in programming style. The unit testing for the proposed system is performed on initialization, Data collection, Genre classification, visualizing the output.

**5.2 Integration Testing**

Developing a programming framework is a sophisticated technique which is used by Integration testing. It solves various issues on dual verification problem and construct program which solves all related problems. Main objective of integration testing is to construct a program structure based on unit testing modules.

As modules of software are divided, testing is performed on each module. Later this separated module is tested as whole set. Here, to rectify errors is difficult as it has different isolated errors. Integration testing is vital in today's IT and software development landscapes, especially when requirements are dynamic and deadlines are tight. Even when each module of the application is unit-tested, some errors may still exist.

To identify these errors and ensure that the modules work well together after integration, integration testing is crucial.

**5.2.1 Up down Integration**

The up down integration deals with development of program framework in incremental way. Modules of each program are coordinated in descending order, and it starts with primary module. This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward, beginning with the main program module. Modules that subordinate to the main program module are incorporated into the structure in either a depth first or breadth first manner.

**5.2.2 Bottom-up Integration**

The below table integrated testing table is divided into integrated classes, functions of each class, how test is performed and result generated. This method begins the construction and testing with the modules at the lowest level in the program structure.

Since the modules are integrated from bottom to up, processing required for modules subordinate to a given level is always available. Therefore, in this case the need for stubs is eliminated. It is important to check whether the testing is error free or not for different classes.

**5.3 Validation Testing**

As completion of testing combination is done, writing computer program is put together in on package. Testing approval is described from various perspectives. Hence testing affirms items limits which are sensibly expected.

**5.4 User Acceptance Testing**:

Key factor for success of any system is user acceptance.  User acceptance testing is performed on users, show which will be successful based on user motivation and knowledge. At time of developing and making required changes,

system under consideration along with prospective system, users undergo constant testing. The changes are made regarding to 3 points .User Acceptance Testing is carried out in a separate testing environment. A change, an update, or a new feature is requested and developed. Unit and integration tests are run. All seems to be in order.

**5.4.1 Black Box Testing**

Functional testing is considered as Black box testing and it mainly focus on functional requirements. It is a software technique where the tester could not predict the testing of internal workings.  Programming code is not examined by the tester, and they need not to have further knowledge of the programming other than its specifications. It is a complementary approach uncovering distinct classes of errors which are:

* Error in performance
* Initialization and termination errors
* Missing function
* Errors in interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality which is tested** | **Inputs** | **Testing** | **Observation** |
| Error in Performance | Wrong dataset | Model Accuracy is calculated. | Failed:  Less accurate. |
| Initialization and termination errors. | Wrong Input | Wrong genre provided. | Failed.  Miscellaneous  Genre provided. |

* Errors in objects

**Advantages of black box testing:**

* Tests are unbiased
* Programming knowledge is not required.
* Based on user point of view test is conducted.
* Low chance of false positives
* Tests can be executed by crowd sourced

**5.5 Assurance of Quality**

Quality assurance is testing and analysis of administration element. The main objective is to give knowledge about item quality to administration. Assurance of quality involves:

**5.5.1 Quality Factors**

The main goal of confirmation value is track product quality and observes procedures to enhance programming.

Quality factors are described in following two categories:

* Directly measured factors.
* Indirectly measured factors.

Quality factors mainly focus on following three things:

* Operational attributes
* Experiences capacity changes
* Versatility.
* Effectiveness
* Time duration.

**Summary**

The chapter describes about various testing techniques such as unit testing, integration testing validation testing , output testing etc.

## CHAPTER 6

**SYSTEM REQUIREMENTS**

The system requirements specify features, components and behavior of system which is to be developed. The following sections describe about functional, non-functional, performance related, features and behavior of the solution. This includes the detailed description of the solution to be developed.

**6.1 Functional Requirement**

Functional Requirements are those requirements which show the working and functionality of a system and the expected behavior of a system based on certain situations and inputs. It defines specific functionality of a system. Functional requirements of system are:

* Collecting data set
* Images of various Sides of Rubik’s Cube and different numbers for different colours are required to be obtained.
* Fast running computer that can comb through large amounts of data and create reports at ease and with minimal time.

**6.2 Non-functional Requirement**

Non-functional requirements are not about functionality or behavior of system, but rather are used to specify the capacity of a system. They are more related to properties of system such as quality, reliability, and quick response time. Non- functional requirements come up via customer needs, because of budget, interoperability need such as software and hardware requirement, organizational policies or due to some external factors such as: -

* Basic Operational Requirement
* Organizational Requirement
* Product Requirement
* User Requirement

**6.2.1 Basic Operational Requirement**

The eight primary functions of systems engineering are all performed by the end users, which is the customers. Operational requirements which are given by: -

* **Mission profile or scenario:** It is a map which describes the procedures and leads us to the final goal/ objective. The goal of proposed system is, getting number of clusters for large dataset, selecting centroid and gives less computation time.
* **Performance:** It basically gives system parameters to reach our goal. Parameters for the proposed system are less computational time which is compared to the existing system.
* **Utilization environments:** It enlists the different permutations and combinations a system can be reused in many other applications which gives better cluster quality, as well as gives a new approach to clustering techniques.

**6.2.2 Organizational Requirement**

The Organizational requirement consists of the following types:

* **Process Standards:** To make sure the system is a quality product; IEEE standards have been used during system development .IEEE standards follow a well-defined path from concept to completion, guided by a set of five basic principles: **due process, openness, consensus, balance and right of appeal**. These imperatives ensure fairness and good standards practice during the development cycle, and help validate approved standards.
* **Design Methods:** Design is an important step, on which all other steps in the engineering process are based on. It takes the project from a theoretical idea to an actual product. It gives us the basis of our solution. Because all the steps after designing are based on the design itself, this step affects the quality of the product and is a major player in how the testing and maintenance of a project take place and how successful they are. ​Following the design to the ‘T’ is of utmost importance.
  + 1. **User Requirement**
* The user should be able to have User Interface Window with Visualize Graphics.
* The user should be able to configure with neat GUI all the parameters.
  1. **Resource Requirement**
* **Jupyter Notebook** is a non-profit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". Spun-off from [IPython](https://en.wikipedia.org/wiki/IPython" \o "IPython) in 2014 by Fernando Pérez, Project Jupyter supports execution environments in several dozen languages. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R, and an homage to Galileo’s notebooks recording the discovery of the moons of Jupiter. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, Jupyter Hub, and Jupyter Lab, the next-generation version of Jupyter Notebook. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially refer to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots, and rich media, usually ending with the “. ipynb" extension.
  1. **Hardware Requirements**

Processors : More than inteli3

RAM : 4GB

Storage : 5GB

Standard Devices : Keyboard, monitor, mouse andcamera

* 1. **Software Requirements**

Platform : Windows XP/ 7/10/11/Vista

Language : Python

Simulator : OpenCV

Cloud : GoogleColab

IDE/tool : anaconda

**Summary**

This section tells us about various requirements necessary to run the project on any given system. It also defines the various types of requirements.

# CHAPTER 7

**SYSTEM ANALYSIS**

Analysis is nothing but finding solution to various problems. System analysis is defined as, process in which we get the information about the existing problems, requirements and to solve various problems related to system. Study of feasibility plays vital role in system analysis, which helps in providing goals for development and design.

**7.1 Feasibility Analysis**

Considering initial results, a deeper survey or feasibility study is done on existing results. “FEASIBILITY ANALYSIS” is nothing but meeting the requirement of system, to achieve desired goal, working on proposed system and getting detailed information on resources.

There are 8 steps to be considered for Feasibility Analysis:

* Develop a team for a project and assign leader for that team.
* Identify the strength of proposed work.
* Characterize and recognize qualities of proposed work.
* Identify cost of proposed work and show its performance.
* Show system performance.
* Choose best framework.
* Prepare report and submit to administrative

Three key features involve in “FEASIBILITY ANALYSIS” are:

* Financial Feasibility
* Technical Feasibility
* Public Feasibility
* Scheduling Feasibility
* Legal Feasibility

**7.1.1 FINANCIAL**

Financial feasibility is one key factor of feasibility analysis, which is carried out to scrutinize economic cost of organization, limited fund is endowed by companies for development of system. Hence, the developed project cost is within, budget because of freely available resources and can be accessed by open source.

**7.1.2 TECHNICAL**

Technical feasibility is entailed to scrutinize technical performance of system. System which is building should have less demand. Hence, this gives number high of demands asked by the clients. The proposed system is built in such way that there is no harm for the user as well the MR-SGSO executes very well. The project draws all the technical requirements from open source and resources are used which have higher execution time

**Summary**

The objective of this chapter is to know the proposed system is feasible or not. The chapter mainly describe about the various keys of feasibility analysis i.e., financial, technical, and public.

**CHAPTER 8**

**RESULTS**

# 8.1 Dataset

# The dataset for our application would be each faces of the Rubik’s Cube. Also, each color in the cube is identified with unique number. So, when we show each face of cube to the camera it asks us to show each face of the cube. Like Front, Top, Left, Right, Bottom. So, it input as array of nine numbers in it with each number showing different color. We use Kociemba’s algorithm, and we make use of Open CV and Image processing techniques in this project.

# Screenshot (232)

# Figure 8.1 Diagram of Rubik’s Cube Problem Solver

# 8.2 Final Output

# After performing all the necessary steps the Rubik’s Cube will be solved. Our goal here is to reach the goal in minimum steps possible. The below figures show the steps or methods through which the Rubik’s cube can be solved.

# Arrow marks in the figure shows the direction in which the cubes in Rubik’s cube need to be rotated in order to obtain the final required output.

# The final output for us would be completely solved Rubik’s cube.

# WhatsApp Image 2022-06-20 at 5.10.47 PM

# WhatsApp Image 2022-06-20 at 5.11.04 PM

# WhatsApp Image 2022-06-20 at 5.11.36 PM

# Figure 8.2 Cube Solving-1

# WhatsApp Image 2022-06-20 at 5.12.26 PM

# WhatsApp Image 2022-06-20 at 5.12.44 PM

# WhatsApp Image 2022-06-20 at 5.13.07 PM

# Figure 8.3 Cube Solving-2

# WhatsApp Image 2022-06-20 at 5.13.20 PM

# WhatsApp Image 2022-06-20 at 5.34.58 PM

# WhatsApp Image 2022-06-20 at 5.35.05 PM (1)

# Figure 8.4 Cube Solving-3

# WhatsApp Image 2022-06-20 at 5.39.57 PM

# WhatsApp Image 2022-06-20 at 5.35.35 PM

# WhatsApp Image 2022-06-20 at 5.35.25 PM

# Figure 8.5 Cube Solving-4

# Summary

# The chapter describes about various screenshot of results, step by step process to execute proposed project and advantages of proposed system.

# REFERENCES

### [1] D. Zeng, M. Li, J. Wang, Y. Hou, W. Lu and Z. Huang, "Overview of Rubik’s Cube and Reflections on Its Application in Mechanism," Chinese Journal of Mechanical Engineering, vol. 31,2018.

### [2] E. D. Demaine, M. L. Demaine, S. C. Eisenstat, A. Lubiw and A. Winslow, "Algorithms for Solving Rubik’s Cubes," Algorithms – ESA 2011, vol. 6942, pp. 689-700,2011.

### [3] El-Souranil, N. &. Hauke, S. &. Borschbach and Markus, "An Evolutionary Approach for Solving the Rubik’s Cube Incorporating Exact Methods," Lecture Notes in Computer Science, vol. 6024, pp. 80- 89,2010.

### [4] S.Saeidi, "Solving the Rubik’s Cube Using Simulated Annealing and Genetic Algorithm," International Journal of Education and Management Engineering, vol. 8, 2018.

### [5] S. P. Rohith, A. M. Sharif, S. Jayasankar and M. Harikrishnan, "Autonomous Rubik’s Cube Solver Bot," International Journal of Scientific Research and Engineering Development, vol. 2, no. 3, pp. 146-151,2019.

### [6] D.Gilday, "MindCuber," [Online]. Available: http://mindcuber.com. [Accessed 20 November2020].

### [7] S. P. Rohith, A. Mohamed Sharif, S. Jayasankar, M. Harikrishnan “Autonomous Rubik’s Cube Solver Bot” International Journal of Scientific Research and Engineering Development, Vol.2, Issue 3, (May - June 2019)pp.146-151.

### [8] D. Zeng, M. Li, J. Wang, Y. Hou, W. Lu and Z. Huang “Overview of Rubik’s Cube and Reflections on Its Application in Mechanism” Chinese Journal of Mechanical Engineering,Vol.31, Issue 1, (December 2018): pp.1-12.

### [9] Ayman M. Abo El-Maaty, Magda B. Fayek “Observations on Exploration and Exploitation Effects on Solving Rubik’s Cube Using Evolutionary Strategies” International Computer Engineering Conference, Cairo, Egypt: IEEE, 2017. pp.95-99.

### [10] HassaSamadi, Mohammad Reza Daliri “Solve the Rubik’s Cube with Robot Based on Non- Invasive Brain Computer Interface” Iranian Conference on Intelligent Systems (ICIS), Bam, Iran: IEEE, 2014. pp.4-6.

### [11] D Gebhardt, G Hellings, W H Huang, et al. The cube. New York: Black Dog Leventhal Publishers, 2009.

### [12] T de Castella. The people who are still addicted to The Rubik’s Cube. BBC News Magazine. BBC, 2014 [2018-05-30]. http://www.bbc.com/news/ magazine-27186297.