



- Cubr is a mechanical system capable of solving a 3x3x3 puzzle cube with the intention of speed.
- The solver uses computer vision to detect and map a given cube state into 2D space, creating a cube string, which is passed into our own implementation of the Beginner's Method solving algorithms to find a solution string.
- Our hardware module interfaces the solution string through an Arduino to execute the specified moves to physically solve the puzzle using six bipolar stepper motors.

- Apply creative and innovative technologies to ordinary things
- Show that something that can take days for a human to learn can merely be solved by our robot in seconds.
- Utilize an algorithmic way of thinking, which is pervasive in automation
- Express the power and ability in the decomposition of a complex problem into smaller and simpler subproblems.

The diagram illustrates the hardware and software architecture for a robotic cube solver. It is divided into three main sections: Computer, Arduino Uno Rev3, and Driver-Motor Setup (x6).

Computer Section:

- Cube State Detection Module:** Contains a Logitech C270 Webcam and OpenCV.
- Solver Module:** Contains a Cube object and Solving Algorithms (Beginner's Method and Two-Phase Algorithm).

Arduino Uno Rev3 Section:

- Driver Firmware:** Receives a Solution String from the Computer.
- 12 Digital I/O Pins:** Send two pins for each Driver-Motor setup to the A4988 Stepper Motor Driver.

Driver-Motor Setup (x6) Section:

- A4988 Stepper Motor Driver:** Receives power from a 24V 5A Power Supply and sends power to the NEMA 17 Stepper Motor. It also receives control signals from the Arduino.
- NEMA 17 Stepper Motor:** Receives power from the A4988 Stepper Motor Driver.

Key:

- Newly Designed (Green)
- Pre-existing Implementation (Blue)
- Off The Shelf Part (Yellow)
- Hardware (White)
- Software (Grey)

A 3x3 grid of colored blocks is shown. The blocks are arranged in a 3x3 grid. The colors of the blocks are as follows:

Row	Col 1	Col 2	Col 3
1	Green	Green	Green
2	Green	Blue	Blue
3	Orange	Orange	Orange

A legend in the top left corner shows the color mapping: Green, Blue, and Orange.

1. Change color space to HSV
2. Averaging of HSV values in defined region
3. Classification of color with distance function
4. Create 2D mapping of the cube state

Our implementation solves each layer of the cube by applying a given algorithm to each sublayer and maintaining the previously placed pieces. Solution: 60-100 moves
Steps:

1. White Cross, White Corners
2. Second Layer
3. OLL & PLL (Orient & Permutate Last Layer)

[illegible]

	Color Classification & Thresholding	Naive Beginner's Method	CFOP Method	Our Beginner's Method	Two-Phase Algorithm	Turn Speed (90 degrees)	Turn Delay
Expected Performance	100% success	150 moves	57 moves	100-200 moves	≤ 20 moves	1 sec	2 sec
Actual (Average) Performance	95% success	120 moves	87 moves	84.4 moves	≤ 20 moves	0.065 sec	0.5 sec

Average time to physically solve (Two-Phase): 10.8 seconds