Rubik’s Cube OLL/PLL Trainer

1. Problem Statement: What Are We Trying to Solve?

Congratulations, you’ve solved the Rubik’s Cube! It sits perfectly on the table, each face a single solid color. But for many, this is the end of the road. They followed an online tutorial, completed a few solves, and moved on.

However, for those in the speedcubing community, solving the cube is just the beginning. The focus shifts from merely completing the cube to solving it faster and more efficiently. This requires recognizing complex patterns and memorizing dozens (or even hundreds) of algorithms, an often tedious and overwhelming process.

Our project aims to bridge that gap with a trainer that helps users practice and recognize OLL (Orientation of the Last Layer) and PLL (Permutation of the Last Layer) patterns more efficiently. A brief breakdown of these stages can be seen in the figure below. Instead of scanning algorithm charts and manually matching patterns, our tool will use computer vision to analyze a digital version of the cube, recognize the current configuration, and determine the appropriate algorithm for that stage.

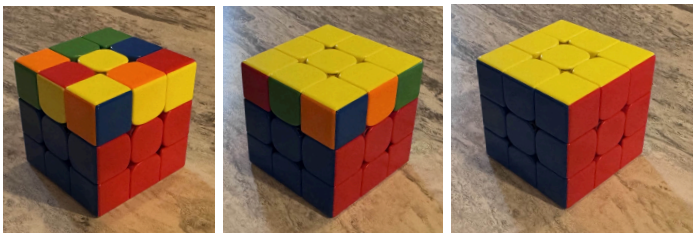

Figure 1. (from left to right) OLL -> PLL -> Solved

2. Why This Problem Matters

Learning algorithms is one of the steepest barriers for aspiring speedcubers. Memorization can be dry, and pattern recognition can feel confusing without a structured practice system. Our trainer aims to make the learning experience more interactive and less time-consuming.

By removing friction from the practice process, we hope to keep cubers more engaged, shorten the learning curve, and potentially increase long-term retention.

3. What Currently Exists and What's Missing

Most existing Rubik's Cube solvers are focused on finding the optimal solution, typically the shortest set of moves to solve the entire cube, after scanning all six faces. While these tools are technically impressive, they're not educational tools and aren't designed for learning or practice.

These "one-shot" solutions are not applicable in real-world speedsolving, where solving is broken down into sequential steps, often using the CFOP method (Cross, F2L, OLL, PLL).

Current tools:

- Don't allow practicing individual OLL/PLL patterns.
- Require static scanning with precisely positioned cube faces.
- Aren't user-friendly for casual or frequent use during practice sessions.

4. Our Approach: A Practice-Focused Enhancement

We're not the first to create a cube solver. Instead, we're repurposing existing technologies with a focus on user experience and practical training.

What sets our approach apart:

- Practice-Oriented: Rather than solving the whole cube, we focus specifically on OLL and PLL, the most algorithm-heavy parts of speedcubing. Additional features such as an automated stopwatch, solve history, and practice scrambles may be added as development progresses.
- Real-Time Algorithms: Using computer vision, our trainer will recognize the current state of a digital cube and determine the appropriate algorithm for that stage.
- Dynamic Interaction: Unlike a lot of existing tools that require static scans, we aim to allow more dynamic interaction, tracking moves in real-time and updating the trainer's output accordingly.
- Optional F2L Expansion: If time permits and the project scope allows, we may expand to include recognition and suggestions for F2L cases, which are typically done intuitively but can benefit from similar pattern-based assistance.

This fills a clear gap: no current solutions provide dedicated OLL/PLL pattern recognition and interactive training tools for speedcubers.

5. Evaluation Strategy

Our project will be evaluated based on:

- Accuracy of color detection from the cube's virtual representation.
- Correct identification of OLL and PLL patterns.
- Responsiveness and user experience of the trainer interface.
- Integration with move tracking to dynamically update the cube state and suggestions.

We plan to demonstrate the tool in a real solve walkthrough, showcasing how it can assist cubers in recognizing and learning OLL/PLL algorithms in context.

6. Timeline & Milestones

Date	Milestone
Oct 1, 2025	Submit project proposal
Mid-Oct	Complete virtual cube generation
Oct 30, 2025	Submit midterm project report
Mid-Nov	Implement pattern detection + tracking
Dec 2, 2025	Present final demo and results
Dec 11, 2025	Publish final project webpage