Almost Real-Time Rubik's Cube Model Reconstruction

Joseph Ng, Brandon Bu
University of California, Santa Barbara
joseph_ng@ucsb.edu
shibu@ucsb.edu

Abstract - **This study introduces an approach to Rubik's Cube state detection, leveraging the strengths of advanced computer vision technologies. With growing interest in automation for educational and recreational applications, precise and efficient detection mechanisms are paramount. Our research utilizes the YOLOv8 model, adapted for dynamic and accurate recognition, in conjunction with a robust HSV-based color detection strategy to accommodate various lighting conditions and Rubik's cube orientations. We further refine our system through enhanced edge detection and noise reduction techniques, thereby improving the fidelity of state analysis. Experimental results demonstrate that our system achieves acceptable performance in nearly real-time detection tasks of being able to process about 16-20 frames a second depending on the hardware. The implications of our findings extend beyond mere puzzle-solving, offering compelling benefits to educational technology sectors that require the manipulation and understanding of complex spatial and color patterns. This work not only propels the field of computational robotics forward but also enriches the toolkit available for cognitive learning processes.**

Keywords - Rubik's Cube, computer vision, YOLOv8, HSV color space, edge detection, real-time processing.

1. INTRODUCTION

The Rubik's Cube has not only become an iconic brain teaser but also a subject of substantial academic interest within the fields of mathematics, computer science, and robotics. The challenge of solving a Rubik's Cube involves recognizing patterns and applying algorithms, making it a suitable model for studies in cognitive processes and automated problem-solving systems. Recent advancements in computer vision and machine learning have opened new avenues for automating such tasks, presenting an opportunity to enhance the capabilities of robotic systems in deciphering complex spatial arrangements and executing precise manipulative actions.

This research is motivated by the dual objectives of advancing educational technology and improving interactive robotics. Rubik's Cube solvers and similar puzzles are increasingly used as educational tools, aiding in the development of spatial awareness, problem-solving skills, and an understanding of algorithmic logic. Professionally, the ability to automate the recognition and solution of such puzzles can lead to significant improvements in the design and function of robots that are required to interact with their environment in a dynamic and adaptive manner.

The primary objective of this study is to develop a robust system capable of detecting the state of a Rubik's Cube using real-time computer vision techniques that are adaptable to various lighting conditions and operational environments. This involves:

1. Enhancing the accuracy and speed of the detection process for recognizing the Rubik's Cube in complex backgrounds
2. Improving the reliability of color detection under different lighting
3. Implementing advanced edge detection and denoising algorithms to ensure precise delineation and recognition of cube segment

2. METHODOLOGY

This section details the methodologies employed in developing our Rubik's Cube state detection system. It includes the selection and adaptation of computer vision models, color detection strategies, and various processing techniques that we considered and implemented. We explored the rationale behind these approaches and selected several to implement in our final system. These technologies were evaluated and selected based on several criteria: 1) processing speed, 2) effectiveness, and 3) robustness in various operational environments.

2.1 YOLOv8 Model Adaptation

The foundation of our approach involves the use of the YOLOv8 object detection model to identify the presence and position of a Rubik's Cube within a video stream or static image. YOLOv8, an evolution of the original You Only Look Once framework, offers significant improvements in detection speed and accuracy, making it well-suited for real-time applications. For this project, the model was retrained on a dataset consisting of hundreds of images of Rubik's Cubes in various settings. These images included diverse backgrounds, from plain single-color surfaces to cluttered, dynamic environments, ensuring the model learns to distinguish the cube in any situational context.

The dataset also included images under different lighting conditions from low indoor light to bright outdoor settings, and cube orientations to ensure the model's robustness and a good accuracy.

In practice, the model operates in real-time, processing input from standard video streams or webcams. As frames are fed into the system, YOLOv8 efficiently identifies potential cubes and assigns each detection a bounding box with an associated confidence score. This score quantifies the likelihood of the bounding box accurately representing a Rubik's Cube, distinguishing between solved and unsolved states based on the arrangement of colors visible on the cube's visible face (*Figure 1*).

Figure 1 - A solved Cube detected by the model with the bounding box

All the detected bounding boxes are critical for the subsequent stages of processing. These bounding boxes are precisely defined to encapsulate the Rubik's Cube, isolating it from any background complexities. This isolation is crucial as it significantly reduces computational load in subsequent steps such as color segmentation and state analysis.

2.2 HSV-Based Color Detection

Following the detection and isolation of the Rubik's Cube within the frame, the subsequent phase involves the meticulous identification of each tile's color. The inherent variability in lighting conditions necessitates a robust approach to color detection, for which the HSV (Hue, Saturation, Value) color model is utilized, offering significant advantages over the traditional RGB model.

The definition of HSV ranges for the colors of the Rubik's Cube is critical to accurately isolating each color in varying lighting. A comprehensive set of images capturing the Rubik's Cube in diverse lighting scenarios is analyzed to establish baseline color values in the HSV space.

This calibration involves manually selecting color samples from images to determine initial HSV thresholds. These initial thresholds are refined through iterative testing, adjusting the hue, saturation, and value ranges to minimize errors in color detection. The range of red and orange need to be carefully adjusted since they share a similar range of hue values. Furthermore, the user's hand and some wood backgrounds have similar HSV ranges as red or orange color.

Figure 2 - Input frame (*left*), visualized composite mask (middle), output frame (*right*) before and after applying HSV-Based Color Detection and Masking.

The detection process involves the binary masking and mask combination. For each color, a binary mask is generated using the OpenCV. The OpenCV function creates a mask where pixels within the specified HSV range are white (indicating presence of that color) and all others are black. Individual masks for each color then are combined to form a composite mask that covers all relevant colors on the cube (*Figure 2*). This composite mask is crucial for segmenting the cube accurately, ensuring that no colors are missed in the detection process.

2.3 Canny Edge Detection

Accurate edge detection is critical for segmenting the Rubik's Cube facelets (aka color tiles) and ensuring precise state recognition. We tested the Canny edge detection algorithm from the OpenCV library. The results from the Canny edge detection algorithm were combined with additional filters and morphological operations to isolate each individual color facelets. This combination proved useful in minimizing false edges and enhancing the clarity of the cube's facelets.

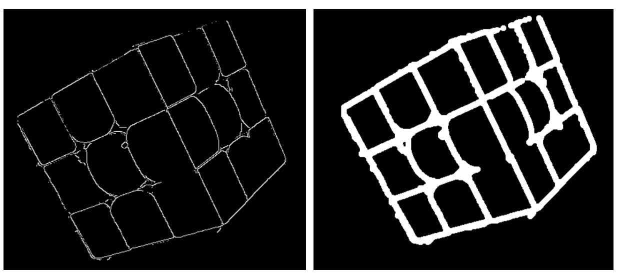

Figure 3 - Canny edge detection (*left*) and morphological dilation + additional filters (*right*)

Applying the processed Canny edges as a subtractive mask to the image isolates each color facelets into its own disjoint regions. We plan on using this data to extract the relative size and orientation information of each facelet color tiles. This information will be useful in correcting errors in which two or more color facelets are incorrectly identified as one continuous region (Figure 4).

Figure 4 - Two orange facelets and two yellow facelets are not separated correctly. The estimated size of facelets that are correctly isolated can be used to cut up the regions that are incorrectly conjoined.

2.4 Denoising Techniques

Noise can significantly impact the accuracy of object detection and recognition tasks. In our task, noise in images arises from various sources, including varying light conditions and shadows, background complexities, and environmental factors (*Figure 5*). Notice that the gray background was not removed entirely in the filtering process. This formed a large noisy region in the bottom right area of the output. Additionally, the shadow casted from the thumb and the similar red-orange color profile of the hand caused two outlier spots in the lower left region.

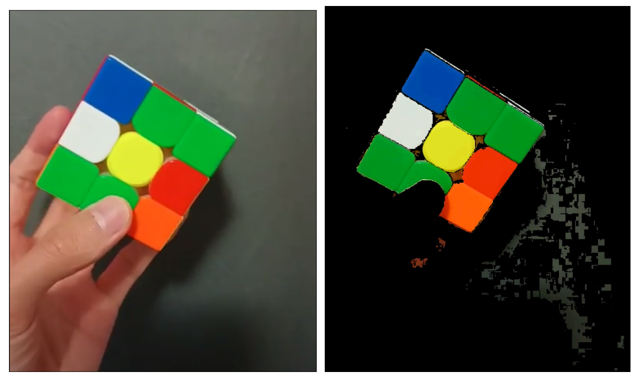

Figure 5 - Input frame (*left*) and noisy output frame (*right*)

To address this challenge, we implemented a denoising strategy to improve the quality of the input images and enhance the performance of our Rubik's Cube state detection system.

The denoising algorithm consists of two stages:

1. Outlier removal using morphological transformations
2. Biggest continuous region selection

In order to remove small outlier regions in the image, such as the two red spots in *Figure 5*, morphological transformations were used. Morphological erosion was used to remove smaller imperfections. Then a dilation of the same scale was used to restore the region to its original size. This combination of transformations are also known as morphological openings.

For larger regions of noise, a large morphological erosion was needed. If the region of noise was not completely removed, the dilation transformation will enlarge and restore the noise to its original size, thus proving it to be useless. However, too large of an erosion factor degrades details from the color mask required for the dilation to restore back to its original shape. Therefore, pure morphological techniques proved to be unsatisfactory.

Instead, these regions of noise can be eliminated by only keeping the largest continuous region in the color mask and discarding the other disjointed regions. This operation was successful in removing large regions of noise similar to the noise in *Figure 5.* However, it was not surprising that this denoising technique has the failure case in which the noise regions are connected with the main cube color mask (*Figure 6* + *7*).

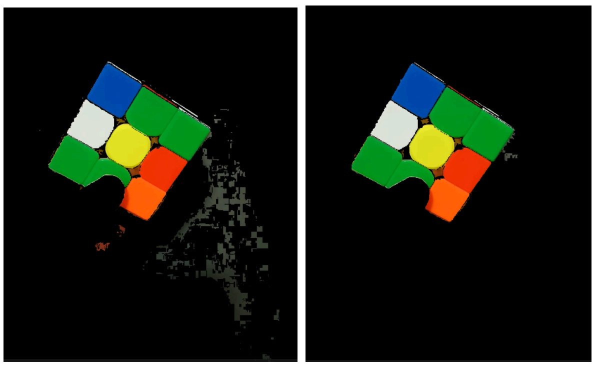

Figure 6 - Noisy input frame (*left*), denoised output frame (*right*)

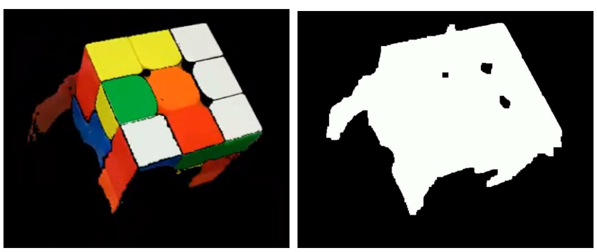

Figure 7 - The denoising algorithm was unable to remove the red region caused by the user's fingers because the color mask was one continuous color region.

4. RESULT

This section presents the results obtained from the implementation of the proposed Rubik's Cube state detection system, which integrates YOLOv8 for object detection, HSV-based color detection, Canny edge detection, and various noise reduction techniques. The performance was evaluated based on accuracy, speed, and robustness under varying lighting and background conditions.

4.1 Color Detection and Masking Performance

The application of the HSV-based color detection approach was particularly crucial for identifying the correct color configuration of the cube. The post-processing images reveal that the colors of each tile are vivid and accurately matched with their real-world counterparts (*Figure 2*). The masks applied effectively isolated each color, even in cases where similar hues could potentially lead to confusion. However, it encounters challenges under extreme lighting or shading conditions. These conditions can lead to saturation or shadows that obscure the true colors of the tiles and in turn affects the accuracy of the mask outputs.

The adaptation of the HSV system for color detection generally provides reliable segmentation of the cube colors across a range of environments. However, as depicted in Figure 8, certain extreme lighting conditions, such as direct light or strong shadows, can introduce high contrast and brightness that skew the perceived hues and saturations, causing the erroneous or incomplete masks (*Figure 8*).

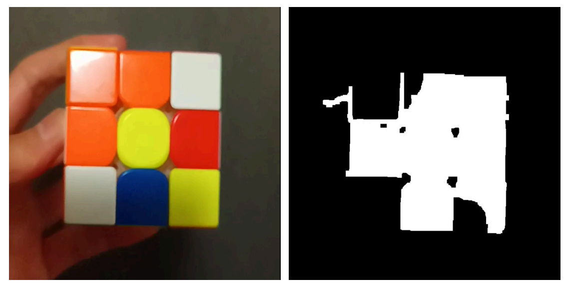

Figure 8 - Input frame with some lighting and shading (*left*), mask output (*right*)

4.2 Processing Speed and Real-Time Capability

An essential aspect of evaluating the Rubik's Cube detection system's effectiveness is its processing speed and ability to operate in real-time. Achieving optimal processing speed is crucial, especially for potential real-time applications. Initially, the project aimed to implement real-time detection using a webcam. However, early experiments revealed challenges in maintaining consistent processing speeds necessary for live interactions. Factors such as variable frame rate from the webcam and the computational load of processing without prior optimization led to significant delays. Therefore the decision was made to utilize pre-recorded videos. This approach allowed for a controlled environment to better measure and optimize the system's performance and capabilities without the variables introduced by live video.

The system was tested using an 8 second video of a Rubik's cube being manipulated in various orientations and lighting conditions. The total processing time for this video ranges from 12 - 15 seconds. This indicates that the system processes at a rate slightly slower than the real-time (16-20 fps). The most time-intensive steps were identified as the color detection and masking processes, particularly when dealing with frames that featured high color variability and complex background.

5. LIMITATIONS AND FUTURE WORK

This research has made significant strides in the field of Rubik's Cube state detection using advanced computer vision techniques. Yet it is not without its limitations that pave way for future improvements. One notable challenge lies in the dynamic HSV color range thresholding; while robust across many lighting conditions, the system still relies on manual calibration for extreme scenarios. This necessity limits the system's autonomy, as variations in ambient lighting not accounted for during calibration may lead to color detection inaccuracies, affecting the overall system performance. Additionally, the system's reliance on color detection for identifying cube stickers is prone to errors from shadows and reflections that mimic color properties, which can result in incorrect cube state outputs.

Looking to the future, there are several promising directions for research and development. First, the development of an adaptive algorithm that can dynamically adjust color thresholds in real-time based on continuous environmental feedback could reduce the need for pre-calibration, thereby enhancing robustness and accuracy in a wider range of lighting conditions. Such improvements would promote greater autonomy in color detection.

While our color-centric approach to detecting cube color shows promising results, future research could explore different trajectories of techniques, such as integrating machine learning algorithms that focus on object model tracking. The neural 6 degrees of freedom tracking and 3D reconstruction algorithm by Wen et al. from Nvidia known as BundleSDF promise accurate model reconstruction at interactive speeds of 10 Hz [1]. Model position and orientation tracking through such advanced algorithms could significantly enhance our system's robustness and accuracy by eliminating the reliance on color masking. Additionally, it would be interesting to explore the tradeoffs in terms of speed, accuracy, and robustness between traditional computer vision color masking solutions and machine learning object model tracking techniques.

Moreover, integrating the Rubik's Cube detection system into XR (extended reality) and VR (virtual reality) environments could revolutionize how users interact with educational content, allowing for immersive experiences where users can virtually manipulate a Rubik's Cube. This integration would significantly enhance learning experiences and cognitive training methodologies.

Furthermore, an exciting avenue for future development is the application of the Rubik's Cube detection system within extended reality (XR) and virtual reality (VR) environments. This integration aims to create an immersive educational platform that not only teaches users how to solve the Rubik's Cube but also enhances their spatial reasoning and problem-solving skills. By employing VR technology, users can interact with a virtual Rubik's Cube in a three-dimensional space, receiving real-time feedback and guidance generated by our advanced detection system. Such an interactive setup would allow users to visualize the cube's state from multiple angles and practice different solving strategies under guided conditions. Moreover, integrating adaptive learning algorithms could tailor the difficulty and teaching strategies based on the user's progress and learning style. This personalized learning experience has the potential to revolutionize educational methods for puzzles and other cognitive tasks, making learning more engaging and effective. The development of this VR-based system would require close collaboration between computer vision experts, educational technologists, and cognitive psychologists to ensure that the platform is not only technologically robust but also pedagogically sound.

6. CONCLUSION

This study presents a comprehensive approach to Rubik's Cube state detection by leveraging advanced computer vision technologies. With the growing interest in automation for educational and recreational applications, precise and efficient detection mechanisms are paramount. Our research utilizes the YOLOv8 model, adapted for dynamic and accurate recognition, in conjunction with a HSV-based color detection strategy to accommodate various lighting conditions and Rubik's Cube orientations. We further refined our system through enhanced edge detection and noise reduction techniques, thereby improving the fidelity of state analysis.

Experimental results demonstrate that our system achieves acceptable performance in nearly real-time detection tasks, capable of processing about 16-20 frames per second depending on the hardware. The implications of our findings extend beyond mere puzzle-solving, offering compelling benefits to educational technology sectors that require the manipulation and understanding of complex spatial and color patterns. This work not only propels the field of computational robotics forward but also enriches the toolkit available for cognitive learning processes.

Our methodologies ensure that the system maintains high accuracy and adaptability under diverse conditions, highlighting the potential for broader applications. The combination of YOLOv8 for object detection, HSV color space for robust color detection, and various denoising and edge detection techniques has proven effective in achieving the desired performance metrics.

Looking ahead, further improvements could include dynamic HSV thresholding to enhance robustness in varying lighting conditions and integrating the system into XR and VR environments for immersive educational experiences. These advancements would further solidify the system's utility in both educational and professional robotics contexts, pushing the boundaries of what automated systems can achieve in interactive and adaptive learning environments.

References

[1] Wen, B., Tremblay, J., Blukis, V., Tyree, S., Muller, T., Evans, A., Fox, D., Kautz, J., & Birchfield, S. (2023). BundleSDF: Neural 6-DOF Tracking and 3D Reconstruction of Unknown Objects. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2303.14158

[2] Hkociemba. (n.d.). GitHub - hkociemba/CubeExplorer: A program about Rubik's Cube with a lot of features. GitHub. https://github.com/hkociemba/CubeExplorer

[3] Wang, M., Wang, Y., Shi, Q., & Li, S. (2022, November 1). Research on state recognition of Rubik's Cube based on monocular Vision. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=10218571&ref=&tag=1

[4] Reis, Dillon, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. "Real-Time Flying Object Detection with Yolov8." arXiv.org, May 22, 2024. https://arxiv.org/abs/2305.09972.

[5] OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., & Zhang, L. (2019, October 16). Solving Rubik's Cube with a Robot Hand. arXiv.org. https://arxiv.org/abs/1910.07113