ColorSense: Color Vision Deficiency Diagram Converter

Shanmu Wang
UCLA
Los Angeles, USA
shanmu@ucla.edu

Tiffany Zhou
UCLA
Los Angeles, USA
tiffanyzhoujh23@ucla.edu

Xiaojie Zhou
UCLA
Los Angeles, USA
xiaojiez6@ucla.edu

ABSTRACT

Color blindness affects approximately 4% of population worldwide, making it challenging for people to interpret content that rely on color differentiation in real life. In college, students with color vision deficiency can struggle with understanding course material that requires color differentiation, especially diagrams and graphs. Teaching assistants (TAs) can unknowingly create diagrams that are difficult for colorblind students to understand, contributing to a persistent accessibility gap. In this paper, we propose ColorSense, a web-based tool designed to address this issue by enabling TAs to convert diagrams into more accessible forms. Users can upload a diagram, select the type of color blindness they wish to accommodate, and receive a modified version with adjusted color mapping and added patterns for improved differentiation. The tool currently supports bar plots, applying both color mapping and hatched patterns to the original graph. ColorSense aims to simplify the process of creating accessible teaching materials, ultimately improving the learning experience for colorblind students.

KEYWORDS

Accessibility, Color Vision Deficiency, Computer Vision

1 INTRODUCTION

Color blindness affects a significant portion of the population, with approximately 1 in 12 men and 1 in 200 women experiencing some form of Color Vision Deficiency (CVD) [3]. In educational settings, visual aids such as bar plots and diagrams play a crucial role in explaining complex concepts. However, students with CVD often struggle to interpret these materials. For example, Figure 1 shows how a diagram can appear differently to individuals with various types of color blindness[1], highlighting the challenges in distinguishing colors. Teaching assistants (TAs) may unknowingly create diagrams that rely heavily on color differentiation, making it difficult for colorblind students to follow along. As a result, these students are left to find their own workarounds, such as asking peers for clarification or simply avoiding raising the issue.

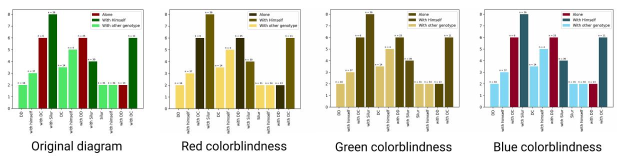

Figure 1: Colorblind Simulations. The original diagram appears differently depending on the viewer's type of color blindness, making it challenging to distinguish between colors.

The challenge arises because TAs are not typically trained in designing accessible teaching materials, and colorblind students may hesitate to speak up about their difficulties. Even if TAs are aware of such an issue, there does not exist a handy tool that can convert the diagrams for them. As a result, the problem persists, creating an unequal learning environment. Existing solutions for improving diagram accessibility are often complex (e.g. requires extra coding knowledge) or require advanced design knowledge, making them impractical for TAs to adopt in day-to-day teaching. Therefore, we make the following *problem statement*: TAs can unknowingly create teaching material with diagrams that rely on color differentiation, making them difficult for colorblind students to interpret. We want to make a tool that makes it simple to convert diagrams into accessible forms so that we can submerge the accessibility gap where students struggle to understand visual materials efficiently.

To address this issue, we developed ColorSense, a web-based tool that enables TAs to quickly transform diagrams into accessible formats. Users can upload a diagram, specify the type of color blindness (e.g., red, green, blue blindness, or complete color blindness), and receive an adapted version with adjusted color mappings and added patterns for improved differentiation. By streamlining the process of creating accessible diagrams, ColorSense helps bridge the accessibility gap and enhance the learning experience for colorblind students. However, the tool is currently limited to processing bar plots and does not yet support other types of diagrams such as line graphs, pie charts, or scatter plots. Additionally, ColorSense does not account for other visual impairments beyond color blindness, such as low vision or

[1]We use algorithm proposed in [2] to simulate the color deficiency, which is the only one that also models tritanopia reasonably well

Tool	Description	Strengths	Limitations
Anthology Ally for Canvas	Improves the accessibility of digital content, particularly in learning management systems like Canvas.	Identifies missing descriptions in figures or low contrast text in slides; converts course content into different formats (e.g., audio, ePub, HTML) to support diverse learning needs.	Does not focus on color blindness in figures.
DaltonLens	Simulates how colorblind individuals perceive the figures.	Useful for developers and designers creating accessible visuals.	Does not provide feedback and automative adjustments.
Color Oracle	Simulates colorblind vision on-screen to help evaluate accessibility.	Provides real-time simulation for multiple types of color blindness.	Does not provide feedback and automative adjustments.
Microsoft Accessibility Checker	Identifies common accessibility issues in Microsoft Office apps including Outlook, Excel, and PowerPoint.	Detects missing alt. text, contrast issues, and reading order problems.	Does not focus on color blindness in figures.
Viz Palette	Helps design colorblind-friendly palettes.	Allows users to test and refine color schemes for accessibility.	Does not provide automative adjustments.

Table 1: Comparison of accessibility tools for colorblind-friendly design

contrast sensitivity issues. Furthermore, ColorSense does not aim to educate TAs about the existence of color blindness issues—it is designed as a corrective tool rather than an educational one. Future work may address these broader accessibility challenges.

2 RELATED WORK

2.1 Challenges of CVD in Education

Students with CVD struggle with interpreting maps, charts, images, and other color-coded instructional materials, which can hinder their academic performance. Zorn and McMurtrie highlight that most teachers are unaware of color deficiency and make little or no adjustments for affected students, leaving them to navigate these challenges on their own [7]. This lack of accommodation creates a persistent accessibility gap, as students with CVD often face difficulties in analyzing data, reading graphs, and performing science experiments involving color changes (e.g., pH tests, chemical reactions. Stoianov et al. conducted a comprehensive review of the impact of CVD on daily life and education [6]. They found that CVD significantly affects school performance, particularly in subjects that rely on color coding, such as science and geography. Students with CVD often face challenges in reading graphs, distinguishing color patterns, and following color-based instructions. Furthermore, medical students with CVD report difficulties in identifying stained slides and distinguishing between colored indicators in clinical practice, which can affect both their education and future professional performance.

2.2 Addressing CVD in Educational Settings

Meeks et al. propose a Universal Design for Instruction (UDI) approach to address CVD-related barriers in medical education [5]. Their study emphasizes modifying teaching materials by replacing color-coded information with alternative patterns, labels, and grayscale images. For example, histology slides with red and green stains can be replaced with high-contrast grayscale versions, making them more accessible to students with CVD. Additionally, providing extra time during exams and using non-color-based indicators (such as shapes and symbols) has been shown to improve learning outcomes for students with CVD, which has been accommodated in some institutions nowadays.

2.3 Gaps in Existing Solutions

While these studies identify the challenges of CVD in education and propose solutions, most existing strategies require specialized adjustments by educators or institutions. Tools that automate the process of making diagrams accessible remain limited, as summarized in Table 1 In contrast, ColorSense addresses this gap by providing a user-friendly solution that allows TAs to convert diagrams into accessible forms without needing specialized knowledge of CVD or design principles. This approach reduces the burden on both students and educators while improving accessibility in higher education.

3 USER RESEARCH

To ensure the design of ColorSense effectively addresses accessibility challenges in academic materials, we conducted user research with colorblind students, teaching assistants (TAs), and an accessibility expert. Our research aimed to understand the barriers that colorblind students face when engaging with visual materials and to identify practical solutions that TAs would be willing to adopt.

3.1 Research Methodology

We used *semi-structured interviews* as our primary research method. This approach allowed us to gather qualitative insights into the experiences and challenges faced by both students and educators. Participants included:

- Three colorblind students recruited from public forums.
- Seven teaching assistants (TAs) from the UCLA Computer Science department.
- One accessibility expert from the UCLA Center for Accessible Education (CAE).

Interviews were conducted either in person or online, depending on participant preference. Researchers followed a structured interview protocol, ensuring consistency across sessions while allowing participants to elaborate on their experiences.

3.2 Findings

3.2.1 Challenges Faced by Colorblind Students: All student participants reported different levels of difficulty interpreting figures in lecture slides. The most problematic visualizations included bar charts, line graphs, and heatmaps. Students developed various workarounds themselves, such as using phone-based accessibility tools or asking peers for clarification, but these approaches were time-consuming and disruptive to learning.

Notable, none of the students had ever reported their difficulties to a TA or lecturer, as they do not want to be perceived as a burden. This highlights a significant *awareness gap*: TAs may assume that accessibility is not an issue simply because students do not raise concerns.

3.2.2 Teaching Assistants' Awareness and Constraints: Our interviews with TAs revealed that most were aware that colorblind students might struggle with certain visuals, with only one participant actively checked their slides for accessibility before lectures. The primary reasons for neglecting accessibility included:

- Lack of awareness of available tools for checking and improving color accessibility.
- Time constraints, with most TAs willing to check slides for any potential accessibility issues within a five-minute preparation window.
- No direct feedback from students, leading to the assumption that existing materials were adequate.

Furthermore, TAs expressed strong interest in a tool that could provide **automated feedback** and **quickly generate accessible versions of diagrams**. The most requested features included:

(1) Pattern and texture suggestions to complement color differentiation.
(2) A streamlined workflow, minimizing additional effort during slide preparation.

3.2.3 Expert Insights on Existing Solutions: Our discussion with an accessibility expert at the Center for Accessible Education (CAE) revealed that some universities manually modify teaching materials for students with disabilities. However, these modifications require specialized software (e.g., Adobe Illustrator for SVG-based adjustments), making them inaccessible to TAs who lack design expertise.

CAE also highlighted the limited automation in current accessibility tools. While some platforms, like Canvas, assess general document accessibility, they do not specifically detect or correct color-related issues in figures. This further reinforced the need for an intuitive, automated solution like ColorSense.

4 DESIGN GOALS

Based on our initial user research results, we identified four key design goals for developing a tool that enhances the accessibility of educational materials for colorblind students. These goals, informed by research and instructor feedback, focus on making accessibility improvements straightforward, flexible, and effective.

Design Goal 1: Offer Flexible Adaptation for Instructors: Since instructors use diverse tools and methods to create teaching materials, the tool must accommodate different workflows and preferences. While some may prefer automated fixes, others might want greater control over modifications. The tool should support both approaches by providing automated recommendations alongside manual adjustment options.

Most existing accessibility tools follow a rigid, one-size-fits-all approach, which can be limiting. Instead, this tool should offer tailored guidance based on the instructor's platform—whether they use PowerPoint, LaTeX, or a programming language like Python. If an instructor opts for manual adjustments, the tool should provide targeted recommendations specific to their workflow. By allowing customization, the tool will better integrate into diverse teaching styles, fostering broader adoption.

Design Goal 2: Prioritize Accessibility-First Design: Many instructors assume their materials are accessible without recognizing the barriers certain color choices create. Small decisions—such as using red and green together or relying solely on color to convey meaning—can significantly impact students with color vision deficiencies.

To ensure accessibility is a fundamental consideration rather than an afterthought, the tool should proactively check for issues like poor contrast and color-dependent information, providing clear, actionable suggestions. Instead of simply flagging problems, it should recommend alternative color palettes, patterns, or labeling methods that enhance inclusivity. A preview mode simulating how a colorblind student perceives the material can further enhance awareness. By integrating these accessibility checks into the standard workflow, the tool helps instructors prevent issues before they arise.

Design Goal 3: Ensure Efficiency and Ease of Use: Instructors, particularly TAs, often have limited time for extensive accessibility checks. Many existing tools require specialized knowledge or involve extra steps, making them impractical for regular use. To encourage adoption, this tool must be fast, intuitive, and seamlessly integrated into instructors' workflows.

The tool should provide immediate, actionable feedback with minimal effort. Instead of requiring users to navigate complex accessibility guidelines, it should offer a one-click accessibility check that presents recommendations in clear, non-technical language. Compatibility with common platforms like PowerPoint, LaTeX, and Python will further reduce friction, allowing instructors to improve accessibility without disrupting their existing processes. By making accessibility enhancements quick and effortless, the tool promotes routine adoption of best practices.

5 IMPLEMENTATION

5.1 Frontend

The frontend of ColorSense is built using Vue.js, a progressive JavaScript framework known for its flexibility and reactive data binding. As shown in Figure 2, user interface of ColorSense is designed to be intuitive and accessible. The main components include:

- File Upload Component – Allows users to upload diagram images directly from their device.
- CVD Type Selector – Allows users to select the type of color blindness they want to accommodate, including general, red blindness, green blindness, blue blindness, or complete color blindness. The default option is set to "general," which is accessible to the broadest audience and suitable when TAs are unsure of the specific type of CVD their students have.
- Preview and Output Component – Displays a preview of the modified diagram and allows users to download the adjusted output.

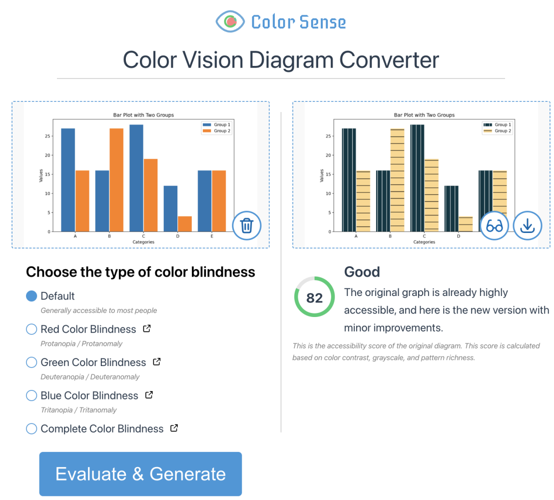

Figure 2: User interface of ColorSense. In the bottom right corner of the result preview are buttons for viewing simulations and downloading.

- Scoring - Displays an accessibility score of the original diagram based on color contrast, grayscale, and pattern richness.
- CVD Simulations - Display the effect of the original diagram as seen by students with different types of colorblindness.

5.2 Backend

5.2.1 Architecture: The backend of our system is designed to provide an efficient and scalable solution for evaluating and enhancing the accessibility of academic visual materials. Built using Flask, a lightweight Python web framework, the backend handles image uploads, accessibility evaluations, and automated modifications to improve readability for individuals with color vision deficiencies. The system follows a modular architecture, ensuring flexibility and easy integration with other platforms. To manage user uploads and generated results efficiently, the backend organizes files into separate directories:

- **user_input/** – Stores uploaded files.
- **output/** – Stores accessibility-enhanced images, such as color-adjusted or pattern-enhanced versions.
- **simulation/** – Stores simulated images representing different types of color blindness.

Each uploaded file is assigned a unique token (UUID), which allows for efficient tracking and retrieval of files without exposing file names directly. This ensures both security and scalability in managing multiple user requests simultaneously. The backend provides a range of API endpoints that allow seamless interaction with the system:

- **/upload** – Accepts image uploads, validates file integrity, and converts non-PNG files to PNG format.
- **/get-preview** – Returns the original uploaded image for preview.
- **/get-simulation** – Generates and serves colorblind simulations (Protanopia, Deuteranopia, and Tritanopia).
- **/get-result** – Applies color adjustments and/or hatching patterns to enhance accessibility.
- **/get-score** – Evaluates the accessibility of the image and provides a numerical score.
- **/clear** – Deletes all stored files associated with a request token to free up storage.

The API endpoints were designed to efficiently manage user interactions, ensuring a smooth and structured workflow for analyzing and enhancing the accessibility of academic visual materials. Each endpoint plays a distinct role, supporting a step-by-step process that takes an uploaded image, evaluates its accessibility, applies necessary modifications, and provides downloadable results. The endpoints are structured to be stateless, meaning each request operates independently, allowing flexibility for frontend interactions without maintaining session states.

5.2.2 Image Processing and Accessibility Enhancements: The backend integrates multiple image processing techniques to enhance accessibility, focusing on improving contrast and reducing reliance on color alone. The simulate_ colorblind module applies transformations that replicate how images appear to individuals with different types of color vision deficiencies. This feature helps educators and content creators understand the challenges faced by colorblind students. The add_hatches_to_bars module enhances bar charts by overlaying distinct hatching patterns, making data points distinguishable without relying solely on color. Additionally, the evaluate_image module assigns an accessibility score based on contrast levels and color differentiation, providing quantitative feedback for improving materials.

To optimize performance, the backend caches intermediate results, such as simulated and modified images, reducing redundant computations and improving response times. The system includes a validation step that ensures only readable, non-corrupted images are processed, reducing the likelihood of errors. Furthermore, the API is designed to handle non-blocking requests, allowing users to initiate accessibility improvements asynchronously, which enhances overall responsiveness.

5.2.3 Containerization and Deployment with Docker: To ensure a streamlined deployment process and eliminate dependency conflicts, the backend is containerized using Docker. This approach allows the system to run in a consistent environment across different platforms, making deployment seamless and efficient. A Dockerfile defines the necessary dependencies and configurations, ensuring that all required libraries and services are bundled within the container.

By running the backend within a Docker container, the tool remains portable and scalable, allowing it to be deployed on cloud-based services or local servers with minimal configuration. Docker also simplifies maintenance by isolating dependencies from the host system, ensuring that updates and patches do not interfere with the underlying infrastructure. This containerized approach facilitates rapid testing, debugging, and scaling, making the system adaptable for larger deployments or future expansions.

5.2.4 Hosting and Deployment Considerations: The hosted and deployed version of the system differs from the local development setup in several key ways to accommodate scalability, persistence, and remote access. In the deployed version, the API endpoints are modified to fit the hosting environment, ensuring proper routing and handling of requests from different network sources. To manage server addresses dynamically, a config.js file is used, allowing the frontend to seamlessly connect to the correct backend URL without hardcoding addresses. This approach enhances flexibility, enabling deployment across different servers without modifying the core application logic. Additionally, the in-memory dictionary (file_store) used in the local version was replaced with an SQLite database to store tokens and file references. This change was necessary because in-memory dictionaries exist only within a single process, meaning they cannot persist across multiple requests in a server environment where different requests may be handled by different worker processes. SQLite provides a lightweight yet persistent solution, ensuring that tokens remain accessible across requests. These modifications allow the system to function reliably in a production environment, supporting multiple concurrent users while maintaining efficient storage and request management.

5.3 Algorithm

5.3.1 Scoring Algorithm: The color accessibility evaluation algorithm assesses the readability and distinguishability of visual materials by analyzing color contrast, grayscale readability, and pattern differentiation, while penalizing problematic color choices such as dark red-green combinations. To achieve this, the system simulates deuteranopia (red-green color blindness) and extracts dominant colors, measuring their perceptual differences in CIE Lab space. Higher scores are assigned when colors remain distinguishable after simulation. Additionally, the algorithm evaluates grayscale contrast to ensure content remains readable even without color and uses edge detection to measure the presence of patterns like hatching, which further enhances accessibility.

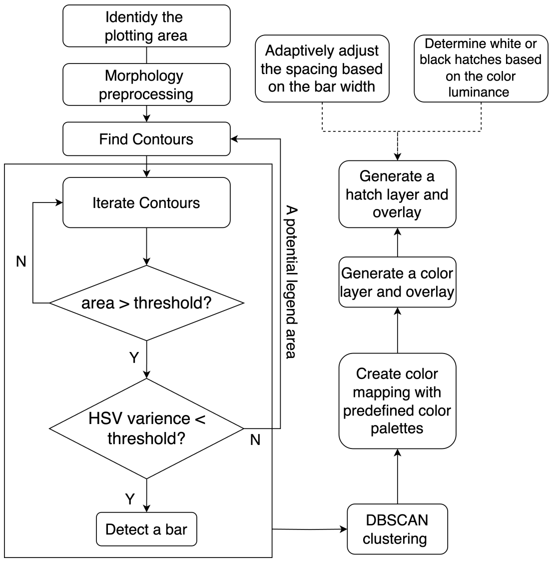

Figure 3: The Diagram of the Computer Vision Processing Algorithm for the Bar Plots.

The final accessibility score is a weighted combination of these factors, with a penalty applied if dark red and dark green are present together. To improve efficiency, tasks such as color simulation, contrast measurement, and pattern detection run in parallel, allowing real-time evaluations. This streamlined approach provides a quick yet comprehensive assessment, helping educators and content creators design more accessible academic materials for individuals with color vision deficiencies.

5.3.2 Computer Vision Processing Algorithm: To improve the accessibility of bar plots for colorblind users, we developed an automated algorithm that detects bars, adjusts their colors, and overlays hatch patterns for better differentiation. The algorithm follows a structured pipeline, as outlined in Figure 3, and is implemented using OpenCV [1] for image processing and DBSCAN clustering [4] for color grouping. Specifically, the algorithm consists of the following steps:

Step 1: Plot Area Identification: The first step in the pipeline involves identifying the plot area within the input image. The algorithm applies Laplacian edge detection to highlight contour boundaries and then extracts the largest rectangular contour as the primary plotting area. To improve accuracy, small margin adjustments are applied around the detected bounding box.

Step 2: Bar Detection and Segmentation: Once the plot area is isolated, the algorithm detects individual bars using adaptive thresholding and morphological operations. By analyzing contour properties and applying filtering criteria (e.g., minimum area threshold and aspect ratio), the algorithm differentiates bars from non-bar elements (such as text or grid lines). The extracted bars are further refined by analyzing their HSV variance, ensuring only solid-colored regions are identified as bars.

Step 3: Color Clustering and Mapping: To enhance color accessibility, the algorithm applies DBSCAN clustering to group bars based on their color properties. This allows bars of similar hues to be categorized and reassigned more distinguishable colors from predefined colorblind-friendly palettes. The palettes, inspired by accessibility research, include options optimized for protanopia, deuteranopia, tritanopia, grayscale, and normal vision.

Step 4: Hatch Pattern Overlay: To further improve accessibility, the algorithm overlays hatch patterns on the detected bars. The choice of hatching is determined adaptively based on:

- **Bar width**, ensuring optimal spacing for visual clarity.
- **Luminance analysis**, where white or black hatches are applied based on the perceived brightness of the bar color.

Five distinct hatch styles are available, including horizontal, vertical, diagonal, cross-hatch, and dot patterns. The algorithm is implemented in Python and is designed to be efficient, allowing for real-time processing of bar plots.

6 EVALUATION

To assess the usability and effectiveness of ColorSense, we conducted a user evaluation involving 10 participants, including TAs and students with or without color vision deficiencies. The evaluation aimed to measure how well the tool improves accessibility in academic materials and identify areas for enhancement.

6.1 Evaluation Methodology

We employed a mixed-methods approach, combining quantitative metrics (e.g., task completion time, usability ratings) with qualitative feedback (e.g., open-ended survey responses). The evaluation followed a structured protocol:

(1) Participants were introduced to the tool and given a scenario-based task, simulating the process of adjusting an academic figure for accessibility.
(2) They completed three conversion tasks, during which we recorded task completion time, interaction behavior, and usability challenges.
(3) Afterward, participants rated their experience on a 5-point Likert scale and provided open-ended feedback on the tool's features and potential improvements.

The participants included 10 individuals, with a mix of TAs and students, some of whom had color vision deficiencies. This diversity allowed us to gather insights from different perspectives and experiences.

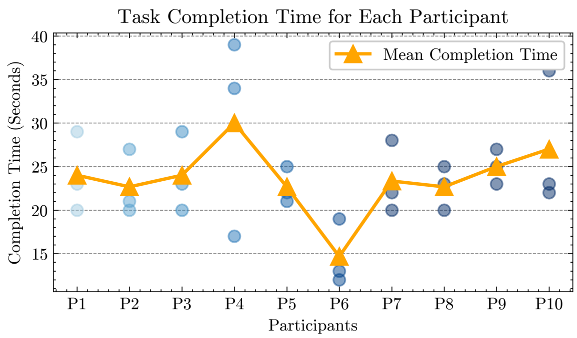

Figure 4: Task completion time per participant. The orange line represents the mean completion time for each participant, which is under 30 seconds.

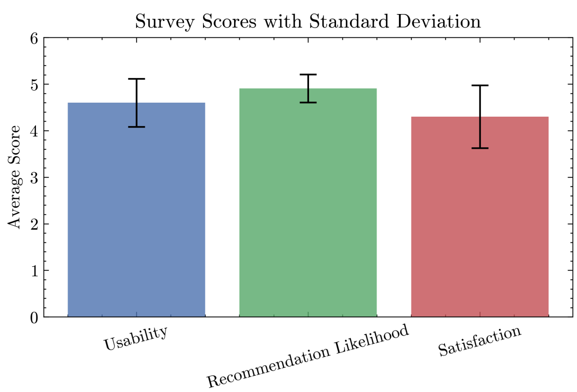

Figure 5: Overall User Ratings, which shows high scores in usability, recommendation likelihood, and satisfaction.

6.2 Quantitative Results and Findings

We present the results of the evaluation, focusing on key metrics and qualitative feedback.

6.2.1 Task Completion Time and Learnability: Most of participants demonstrated a steady improvement in task completion time, with the average time per diagram being under **30 seconds**. Repeated use reduced completion time, indicating that the tool is intuitive and easy to learn. Figure 4 illustrates the task completion time for each participant.

6.2.2 Usability Ratings: We conducted a post-task survey to assess participants' perceptions of the tool's usability, their recommendation likelihood, and their satisfaction on ColorSense. Participants rated the tool on a scale of 1 to 5, with 5 being the highest rating. As shown in Figure 5 Participants rated ColorSense highly across key usability dimensions:

- **Usability**: 4.6 / 5
- **Recommendation Likelihood**: 4.9 / 5
- **Satisfaction**: 4.3 / 5

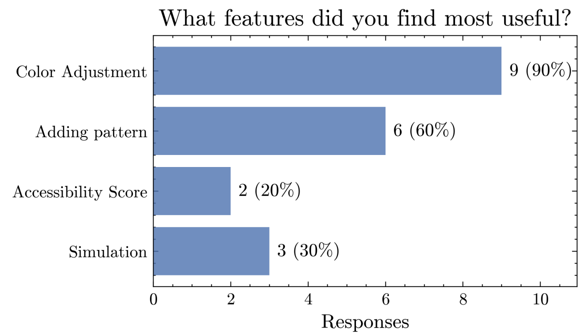

Figure 6: User Feedback on the Most Useful Features in ColorSense.

6.2.3 Feature Preferences: As shown in Figure 6, participants found Color Adjustment (90%) and Pattern Overlay (60%) to be the most valuable features, highlighting their importance in making academic materials accessible.

Additionally, some participants appreciated Accessibility Scoring (20%) and Simulation Mode (30%), which helped them understand how diagrams appear to individuals with color vision deficiencies.

6.3 Qualitative Results: User Feedback and Key Improvements

The post-task survey included open-ended questions, allowing participants to provide qualitative feedback on their experiences with ColorSense. The feedback highlighted both strengths and areas for improvement. Specifically, participants mentioned the following strengths:

- **Intuitive Interface:** The tool's interface was easy to navigate, with clear instructions and visual cues.
- **Effective Color Mapping:** The color mapping feature was praised for its effectiveness in enhancing accessibility.
- **Pattern Overlays:** Participants appreciated the ability to add patterns to color-coded elements, making them distinguishable for individuals with color vision deficiencies.
- **Learnability:** Participants reported a short learning curve, with most being able to complete tasks quickly after initial use.

However, participants also provided valuable suggestions for our future improvement:

- **Batch Processing:** Multiple participants requested the ability to process multiple images simultaneously, which would save time and effort.
- **Mobile Usability:** Some participants expressed interest in mobile-friendly support, which could enhance accessibility for users on different devices.

7 DISCUSSION AND CONCLUSION

The evaluation of ColorSense demonstrated its effectiveness in enhancing the accessibility of academic visual materials. The tool received high usability ratings, with participants particularly valuing its color remapping and pattern overlay features. The steady decrease in task completion time indicates an intuitive interface with a minimal learning curve. However, feedback also highlighted areas for improvement, such as the lack of batch processing and limited mobile compatibility. While ColorSense streamlines accessibility modifications, it remains a corrective tool rather than an educational resource for instructors on best accessibility practices.

Compared to existing solutions, ColorSense stands out by providing automated, user-friendly modifications rather than just identifying accessibility issues. Unlike tools that require manual adjustments or specialized design knowledge, ColorSense enables quick, effective transformations suited for TAs with limited time. Despite its advantages, the tool currently supports only bar plots and does not account for other visual impairments beyond color blindness, indicating directions for future development.

Future work will focus on expanding support for additional visualization types such as line graphs and heatmaps, enhancing system integration with learning platforms, and refining usability with batch processing and mobile-friendly features. These improvements will further reduce barriers to accessibility in education, fostering a more inclusive learning environment.

REFERENCES

[1] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[2] Hans Brettel, Françoise Viénot, and John D Mollon. 1997. Computerized simulation of color appearance for dichromats. *Journal of the Optical Society of America A* 14, 10 (1997), 2647–2655.

[3] Colour Blind Awareness. n.d.. Colour Blindness Awareness. https://www.colourblindawareness.org/colour-blindness/ Accessed: 20 March 2025.

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, Vol. 96. 226–231.

[5] Lisa M Meeks, Neera R Jain, and Kurt R Herzer. 2016. Universal Design: Supporting Students with Color Vision Deficiency (CVD) in Medical Education. *Journal of Postsecondary Education and Disability* 29, 3 (2016), 303–309.

[6] Maristela Stoianov, Mateus Silva de Oliveira, Mariana Cristina Lobato dos Santos Ribeiro Silva, Matheus Henrique Ferreira, Igor de Oliveira Marques, and Mirella Gualtieri. 2019. The impacts of abnormal color vision on people's life: an integrative review. *Quality of Life Research* 28 (2019), 855–862.

[7] Tiffany Zorn and Deborah McMurtrie. 2019. Color deficiency within the classroom. *South Carolina Association for Middle Level Education* (2019), 21.