

Problem Statement and Goals

Bridging Gaps: AI for Diagram Accessibility

Team 22, Reading4All
Nawaal Fatima
Dhruv Sardana
Fiza Sehar
Moly Mikhail
Casey Francine Bulaclac

Table 1: Revision History

Date	Developer(s)	Change
September 17th 2025	Fiza Sehar	First draft of the document
November 3rd 2025	Fiza Sehar	Added metrics for goals and stretch goals

1 Problem Statement

1.1 Problem

Students with disabilities face significant barriers to equitable education, particularly when learning materials rely heavily on technical diagrams. These diagrams are often distributed as static images, making them inaccessible to assistive technologies such as screen readers. The lack of detailed and consistent alternative (alt) text prevents students from independently understanding key visual information, leading to unequal learning experiences in STEM courses. Currently, creating descriptive alt text for such diagrams is a time-consuming and inconsistently applied process. This results in major gaps in accessibility compliance and limits the inclusiveness of post-secondary learning environments. There is a clear need for a scalable and reliable method of ensuring that all technical diagrams are accompanied by clear, accurate, and accessible descriptions that support independent learning for students with disabilities.

1.2 Inputs and Outputs

Inputs: Technical diagrams/images requiring alternative (alt) text descriptions.

Outputs: Automatically generated descriptive alt text suitable for screen readers.

1.3 Stakeholders

Direct Stakeholders

1. **Students with Disabilities:** Primary beneficiaries who will gain access to previously inaccessible technical diagrams.
2. **Instructors and TAs:** Use the tool to generate accurate alt text for course materials.
3. **Accessibility Services Staff:** Ensure that generated descriptions meet AODA standards and institutional accessibility requirements.

Indirect Stakeholders

1. **University Administration:** Interested in improving overall accessibility compliance across courses.
2. **Curriculum Designers:** Can integrate the tool into course development workflows for consistent accessibility.
3. **Assistive Technology Vendors:** Benefit from improved compatibility and user experience when screen readers are used with accessible diagrams.

1.4 Environment

Development Frameworks and Tools:

1. *GitHub* will be used for version control and collaboration.
2. *Visual Studio Code* will be used as the **IDE (Integrated Development Environment)** for development.
3. *Python and Machine Learning Libraries* will be used to implement and train the alt text generation model.

2 Goals

Our overarching goal is to make technical diagrams accessible by generating high-quality, readable alternative text. We will achieve this through the following measurable and outcome-focused goals:

1. **Generate Clear and Concise Descriptions:** Produce alt text that accurately captures key diagram details while remaining easy to read and understand.
Metric: Clarity and readability will be evaluated through user feedback and accessibility testing to ensure descriptions are easy to follow.
2. **Improve User Satisfaction and Learning Outcomes:** Enhance comprehension and satisfaction for students using generated alt text.
Metric: Measured through surveys.
3. **Develop a Browser Extension:** Deliver an intuitive browser extension that allows users to generate alt text directly within their workflow.
Metric: Success will be determined through prototype testing and consistent performance across multiple browsers.
4. **Implement Accessible Hotkeys:** Allow users to navigate, review, and switch between generated descriptions using customizable keyboard shortcuts.
Metric: Evaluated based on usability testing and positive user feedback on efficiency and ease of use.

3 Stretch Goals

1. **Multi-Language Support:** Extend the alt text generation model to support additional languages, improving accessibility for multilingual learners.
Metric: Verified through review by bilingual users to ensure clarity and contextual accuracy.
2. **Batch Processing:** Enable users to process multiple diagrams or documents at once for greater efficiency.
Metric: Assessed through functional testing to confirm reliable performance and maintained quality when handling multiple files.

4 Extras

The overall challenge level of this project is moderate. While developing an ML-based system for generating alt text involves model design, training, and evaluation, there are well-established frameworks and research papers that guide these processes, making them approachable for our team. Our collective experience with Python and machine learning from coursework provides a solid foundation for completing this work successfully. The project will require balancing technical development with accessibility considerations, but its scope remains feasible within the given timeline.

To strengthen the project, we will include two additional deliverables:

- **Norman's Principles Report:** An evaluation of the tool against Norman's design principles to ensure usability.
- **User Manual:** Comprehensive documentation to guide students in using the tool effectively and to support long-term adoption.

These deliverables will help ensure that the final tool is both usable and well-documented for future use.

Appendix — Reflection

Note: The following reflection represents the collective input of our entire group.

1. *What went well while writing this deliverable?*

Our team collaborated smoothly and was able to create a clear plan of action early on. We held multiple meetings to discuss the problem, inputs, and outputs, as well as the overall vision for what our product should look like. Work was divided efficiently, and we supported each other by sharing ideas and reviewing one another's contributions. Our supervisor provided us with useful information and previously researched data, which was particularly insightful and helped shape the initial proof of concept for our project.

2. *What pain points did you experience during this deliverable, and how did you resolve them?*

We did not struggle with defining the problem itself, as we quickly aligned on the key issue of improving accessibility for technical diagrams. The main challenge we faced was uncertainty about the implementation details, particularly what kind of self-training ML model to use and how we would gather or prepare input data (e.g., whether we would use real diagrams for training). There were still many unknowns about the exact approach, but we had enough foundational information to proceed confidently. Meeting with our supervisor was very helpful in this stage as we were introduced to a data analysis study they had conducted, which provided valuable insight into the ideal length of alt text and highlighted weaknesses in e

3. *How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?*

We intentionally avoided designing an overly complex solution by leveraging existing technology as a foundation. Instead of building everything from scratch, we chose to use a web browser extension framework as our baseline, since extensions are widely used and well-documented, making information and implementation examples easy to find. This allowed us to spend less time reinventing the wheel and more time focusing on the core technical challenge of integrating and training the ML model to generate high-quality alt text. This approach keeps the project achievable within the timeline while still providing significant technical depth through model development and accessibility considerations. By leaving advanced features such as multi-language support and batch processing as stretch goals, we ensured that our scope remains realistic but still challenging enough for a senior design project.