GONZAGA UNIVERSITY

School of Engineering and Applied Science

Center for Engineering Design and Entrepreneurship

PROJECT PLAN

October 12, 2025

Image Processing App

Prepared by:

Zobe Murray

Jacob Jarvis

Junyou Guo

Leonard Jia

Reviewed by:

Dan Lenz
Faculty Project Advisor

Dr. Doug Addleman
Project Sponsor/Liaison

Ben McDonald
Design Advisory Board Member

1 Project Overview

1.1 Project Summary

The fields of neuroscientific and psychological research and computer science have become heavily integrated and provide support for each other through innovation and collaboration. This is true for researching human visual perception, where tools have been created for image analysis and modification, but there is a lack of user-friendly options tailored for use at the Gonzaga Department of Psychology.

Our project will produce an application that allows researchers to easily upload and analyze images by utilizing and adding to models previously created for the Department. The system will examine variables such as image characteristics and image differences to help researchers understand how humans perceive and process visual stimuli. The application will feature a simple user interface and clear documentation will help researchers use the application without computer science expertise.

1.2 Project Objectives

The goal of our project is to develop a user-friendly application that allows cognitive researchers to input, modify, and analyze images by integrating and adding to our existing CNN models. Researchers will also receive documentation within the application to support their use of the application if they are not familiar with our tool. These additions will be reviewed, approved, and adopted by the Gonzaga Department of Psychology.

1.3 Project Stakeholders

- **Development Team:** Jacob Jarvis, Junyou Guo, Leonard Jia, Zobe Murray
- **Project Sponsor:** Dr. Doug Addleman, Psychology Department, Gonzaga University - Serves as the project sponsor, providing domain expertise in psychology research. Will guide scientific approach and ensure the application meets educational and research standards for psychology students and faculty.
- **Faculty Advisor:** Dan Lenz
- **Design Advisory Board:** Ben McDonald
- **Target Communities:**
    - Primary Users - Psychology Students: Undergraduate and graduate psychology students at Gonzaga University and other institutions studying cognitive psychology, perception, and research methods. Students that need hands-on experience with experimental design, data collection, and analysis of visual perception. Application will be a learning tool for understanding visual search research and a platform for conducting their own experiments and projects
    - Secondary Users - Psychology Faculty: Professors teaching courses in cognitive psychology, perception, research methods and experimental psychology who can integrate the application into their curriculum for demonstrations, assignments, and student research projects.

1.4 Project Deliverables

Software Product: A desktop application designed to help analyze images, including a user interface for image comparison and modification, and performance metrics. This will be delivered as a .exe file for users.

User Manual: A comprehensive guide detailing installation, usage, and troubleshooting, provided as a downloadable PDF document.

Developer Documentation: Technical documentation covering code structure and APIs, delivered through an online repository for future maintenance and updates.

Performance Evaluation Results: A report summarizing testing outcomes on image and model analysis, to be submitted to the project sponsor and faculty advisor.

1.5 Project Scope

The Convolutional Neural Networks (CNNs) that will be involved in the backend of the image processing app have been developed and tested. They will be integrated into the design of the full application and are out of scope for this project. CNNs are used conventionally for classifying images but have been modified to output a csv file and heatmap with pairwise similarity comparisons between inputted images.
The in scope tasks for our project include:

- Incorporating the CNNs into a full application
- Further developing the capabilities of the models by creating pathways for retrieving more data from images
- Allowing for image modification
- Building a user interface that researchers can use to upload images and interact with our models and tools.
- Writing documentation to help users find the best way to apply the models to their image sets.

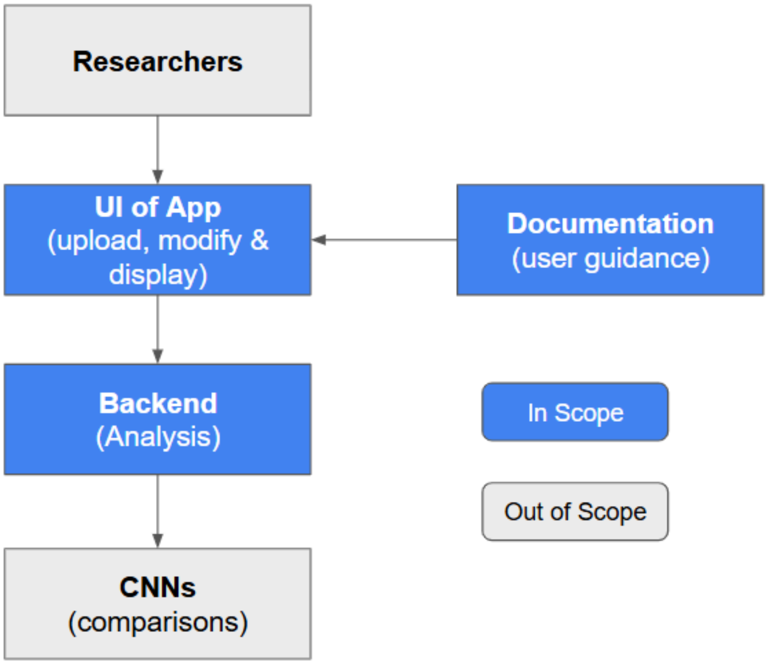


Figure 1: High level context diagram of project scope for image processing app

1.6 Related Work

The **VGG Image Annotator (VIA)** [1] developed by the Visual Geometry Group at Oxford University represents the closest existing system to our proposed application. VIA is a lightweight, browser-based annotation tool that allows users to define regions in images and create textual descriptions of those regions.

Similarities to Our Proposed System:

- **Image-based analysis**: Both systems work with visual content and analyze object locations within images
- **User interaction tracking**: VIA captures user interactions with images through manual annotation processes
- **Region identification**: VIA enables users to identify and annotate specific areas within images, similar to our target object identification

Key Differences:

- **Purpose and scope**: VIA's functionality focuses on manual annotation and labeling, whereas our system provides automated CNN-driven analysis
- **Automation level:** VIA requires manual annotation, whereas our system leverages CNNs for automated image analysis.

[1] https://www.robots.ox.ac.uk/~vgg/software/via/

2 Project Requirements

2.1 Major Features

Our project will produce a user-friendly desktop application that accepts image uploads, performs pairwise similarity comparisons using convolutional neural networks (CNNs) based analysis, and allows users to modify images, visualize results, and compare models. These features will let researchers search for relationships between images, prepare and adapt images for analysis, and interpret their results. We will also provide documentation and user guidance, which will help non-technical users utilize the application to fit their needs.

We considered adding user accounts, long-term storage, and integrating a generative AI feature. However, we opted out of these choices to simplify development and avoid hosting and cost issues. These features may be considered later in the project as stretch goals.

We decided on these project features through discussion in meetings with our advisor and our sponsor. Our sponsor described to us what the target research community values in an application like ours, while our advisor helped us identify what features we should prioritize and that were within our project scope. This allowed us to place features into two categories: target and stretch goals.

Table 1: Major Features

Feature	*Description*
Image Upload and Management	Interface allowing researchers to upload, organize, and manage images or image datasets. Supports common formats (including JPEG, PNG, GIF) with batch upload capabilities.
CNN-Based Image Analysis	Integration of existing Convolutional Neural Networks to analyze uploaded images, generating similarity comparisons, complexity, and visual clutter assessments for psychology research.
Image Modification Tools	Built-in tools allowing researchers to modify images for experimental design, including basic adjustments (brightness, contrast, cropping) and research-specific modifications to test visual perception variables

User Interface for Non-Technical Users	Interface specifically designed for psychology researchers without computer science expertise. Features clear navigation, a minimal learning curve, and research-focused workflows.
Results Visualization and Export	Comprehensive visualization of analysis results, including similarity metrics and charts. Export capabilities for CSV files and visualization to support further research.
Documentation and User Guidance	Integrated help system and comprehensive documentation helping users understand how to apply different models to their specific research questions and image sets.

2.2 Initial Product Backlog

The following requirements are organized using a priority scheme where High indicates features essential for MVP delivery, Medium represents important features for full functionality, and Low indicates nice to have features. Estimates are provided in story points (1-13 scale) based on complexity and development time.

Table 2: Initial Product Backlog

Requirement	*Description*	*Major Feature*	*Priority*	*Estimate*
ImageUpload	System shall allow users to upload individual or batches of images in various formats. Acceptance: single and batch image uploads with format verification and error handling	Image upload and management	High	3

CNNSimilartyAnalysis	System shall implement CNN-based pairwise similarity analysis generating numerical similarity scores (0-1) between uploaded images. Acceptance: similarity matrix output with confidence scores.	CNN-Based Image Analysis	High	8
BasicImageModification	System shall provide basic individual and batch image editing tools including brightness, contrast, and saturation adjustments. Acceptance: slider controls, save/revert	Image Modification Tools	Medium	6
NonTechnicalUI	System shall provide intuitive interface with clear navigation, tooltips, and research-focused workflows requiring minimal technical knowledge. Acceptance: User testing with researchers	User Interface for Non-technical Users	High	5
SimilarityVisualization	System shall generate interactive similarity matrices, and scatter plots for analysis results. Acceptance: viewable visualizations	Results Visualization and export	high	6
UserDocumentation	System shall include comprehensive user guide with tutorials and FAQ. Acceptance: step-by-step tutorials covering all major workflows and troubleshooting guide.	Documentation and User Guidance	Medium	4

2.3 Additional Features

Category	*Feature*	*Description*	*Priority(only compared with Additional Features)*
Technical	Advanced Image Modification Tools	Batch editing with options like noise,blur,rotation,and occlusion commonly used in research	Medium
Technical	Extensible Plugin System	Allow researchers or developers to build plugins that extend models and tools	Medium
UX	Generative AI Function(?)	Utilizing the relevant open sources of AI product improve user experience, such as automatically generate experimental images with controllable variables (e.g., complexity, color, shape)	Medium
UX	Expanded Data Visualization	Provide advanced interactive visualizations such as 3D similarity spaces or response time distributions	High
Teaching& Research	LMS Integration	Integrate with platforms like Canvas,Blackboard or Moodle,enabling instructors to assign work and students to submit experiments directly	Medium

3 Design Considerations

3.1 Initial User Interface Design

The user interface for the Image Processing App is designed with a primary focus on usability for non-technical researchers in psychology. The layout follows a logical, linear workflow that mirrors the research process: from uploading and preparing stimuli, to analyzing them, and finally to interpreting and exporting results. The design emphasizes clarity, minimalism, and intuitive navigation to ensure a shallow learning curve.

The interface is structured around several key screens, each catering to a specific stage of the workflow:

- Main Dashboard / Image Upload Screen: This is the initial landing page. It features a prominent drag-and-drop zone for uploading multiple images, with clear format specifications and feedback. A thumbnail grid of uploaded images provides visual confirmation. A central navigation menu or toolbar gives access to core functionalities: "Upload," "Modify," "Analyze," and "Results."
    - Accommodated Feature: Image Upload and Management.

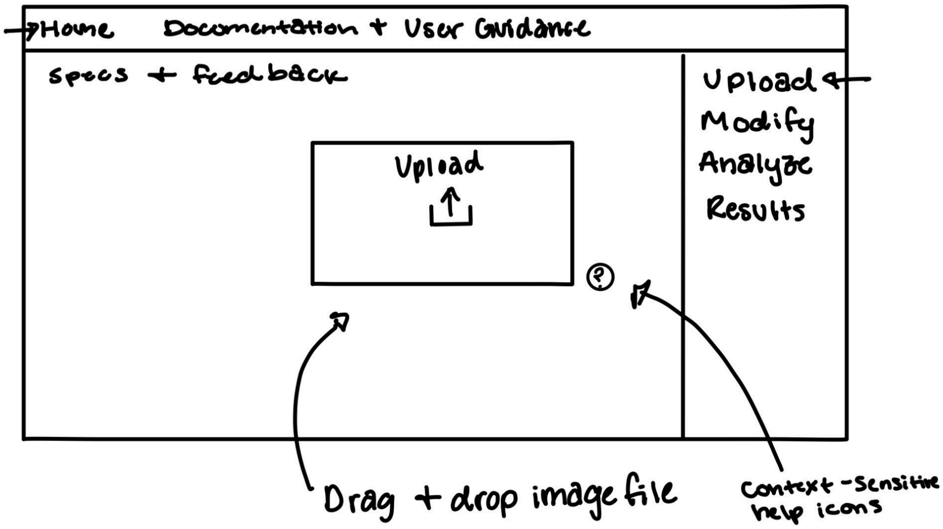


Figure 1: Main dashboard for image processing application

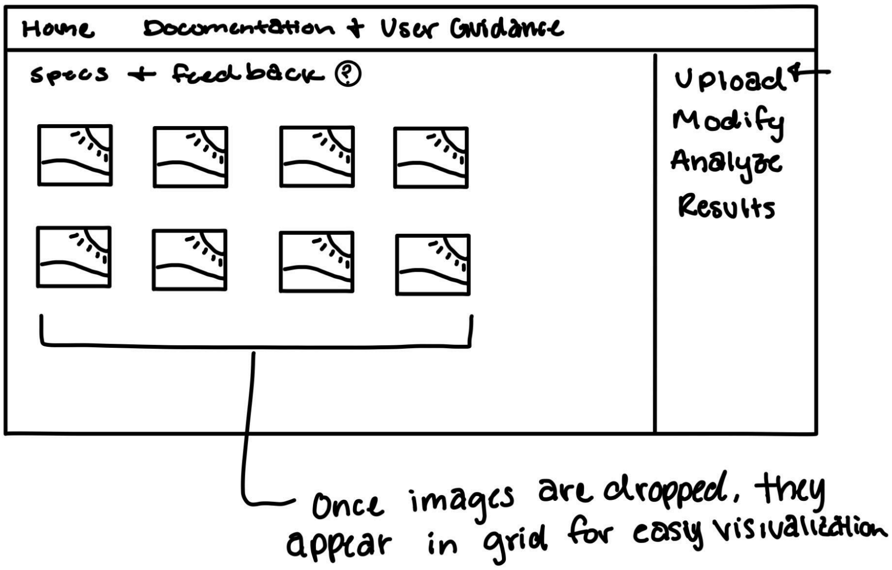

Home
Documentation + User Guidance
specs + feedback
Upload
Modify
Analyze
Results
Once images are dropped, they appear in grid for easy visualization

Figure 2: Image processing application upload screen

- Image Modification Page: Accessed from the main dashboard, this view presents the selected image alongside a set of adjustment tools. Controls for brightness, contrast, and saturation are implemented as sliders with real-time previews. Buttons for "Apply," "Revert," and "Save as New" are clearly visible.
    - Accommodated Feature: Basic Image Modification Tools.

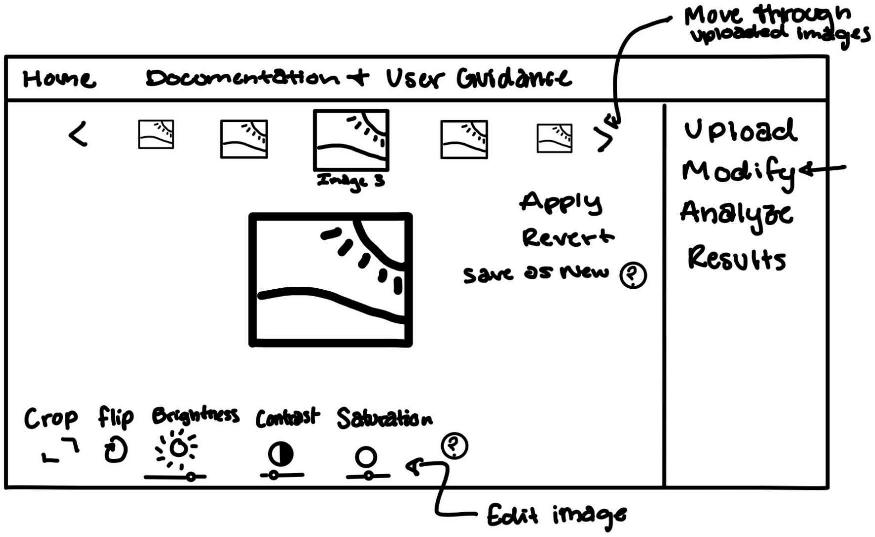


Figure 3: Image modification page with image editing tool bar

- Analysis Configuration & Results Visualization Screen: This screen allows users to select analysis models (e.g., the pre-trained CNN for similarity) and initiate processing. Upon completion, the results are displayed prominently. The centerpiece is an interactive similarity matrix heatmap, where cells can be clicked to reveal detailed pairwise comparisons. Additional visualizations, such as scatter plots, are accessible via tabs. Export buttons for CSV data and high-resolution charts are placed conspicuously.
    - Accommodated Features: CNN-Based Image Analysis, Results Visualization and Export.

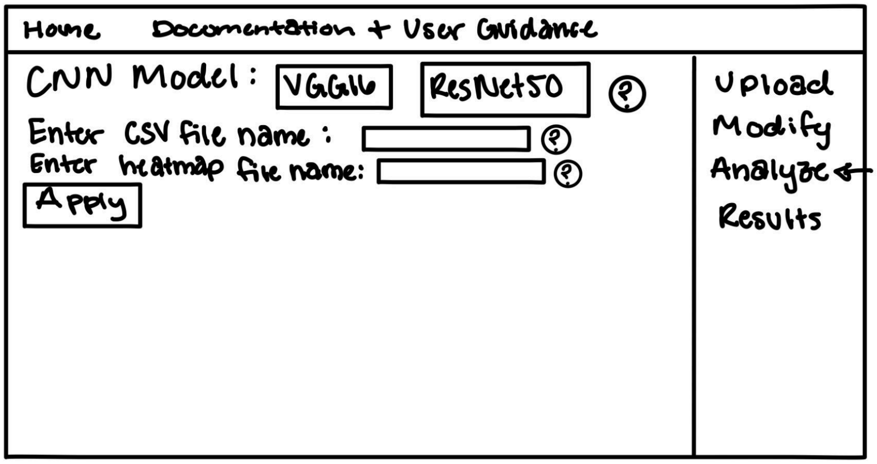


Figure 4: Analysis configuration screen with fields to let user modify the model they use

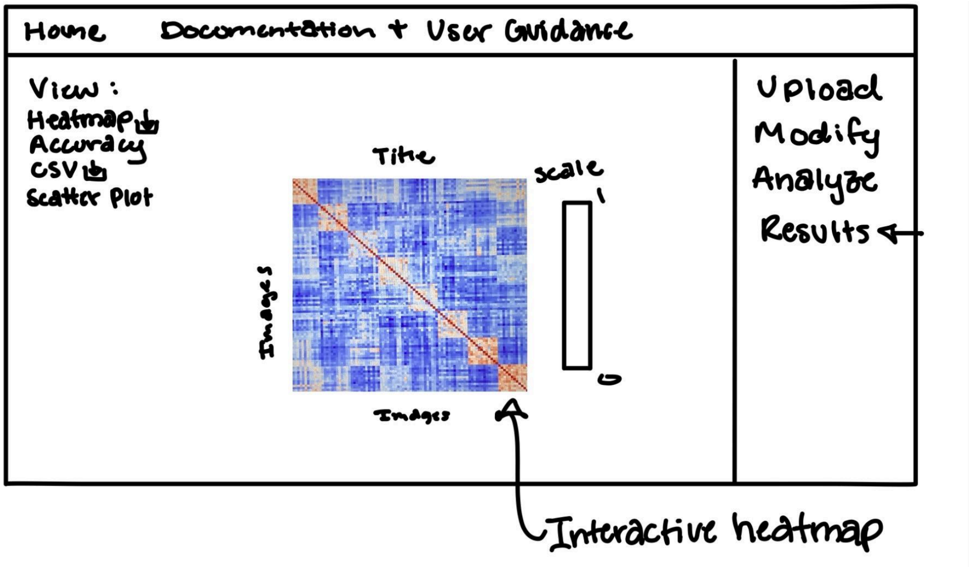


Figure 5: Results visualization screen with interactive heatmap and downloadable files for further and future analysis

- Integrated Documentation: Context-sensitive help icons (e.g., a "?") are located next to complex controls. Clicking them opens a panel with concise explanations and links to the full user manual.
    - Accommodated Feature: Documentation and User Guidance.

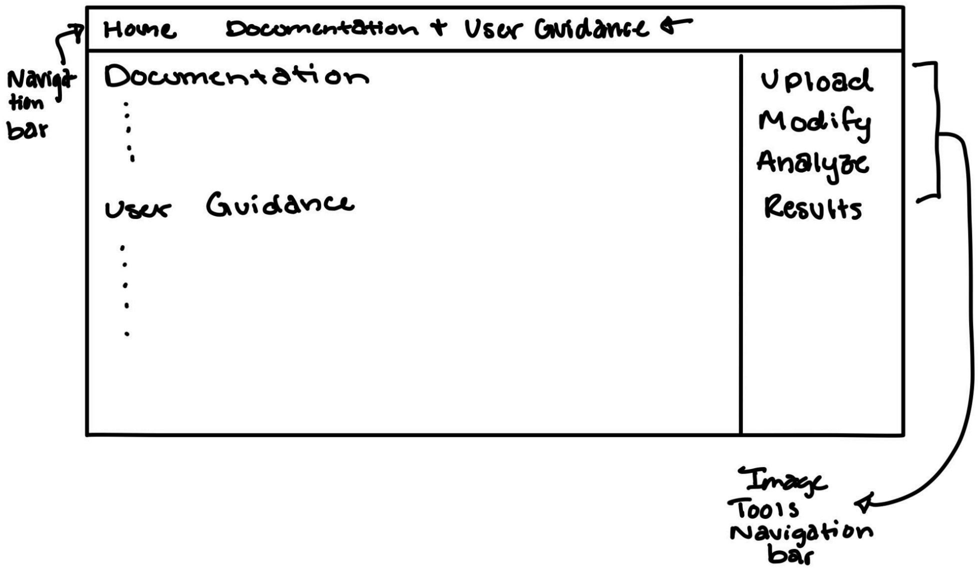


Figure 6: Documentation and user guidance page

Design Process and Vetting:
The initial mock-ups were hand-drawn on an iPad with arrows and notes to provide clarity. The design process was highly iterative and involved continuous feedback from key stakeholders. The mock-ups were first reviewed with our Project Sponsor, Dr. Doug Addleman, to validate that the workflow aligns with the cognitive research processes and terminology used by psychologists. Subsequently, they were presented to our Faculty Advisor, Dan Lenz, and Design Advisory Board member, Ben McDonald, for feedback on technical feasibility and UI/UX best practices. The feedback received focused on simplifying jargon, ensuring the analytical results are presented in a statistically meaningful way, and optimizing the placement of export functions. This vetting process ensured that the design effectively bridges the gap between technical functionality and end-user needs.

3.2 Initial Software Architecture

Three-tier Architecture:

- Presentation Layer: React frontend for non-technical users
- Application Layer: Flask with separate services for upload, analysis, and export
- Data Layer: PostgreSQL for metadata, file storage for images

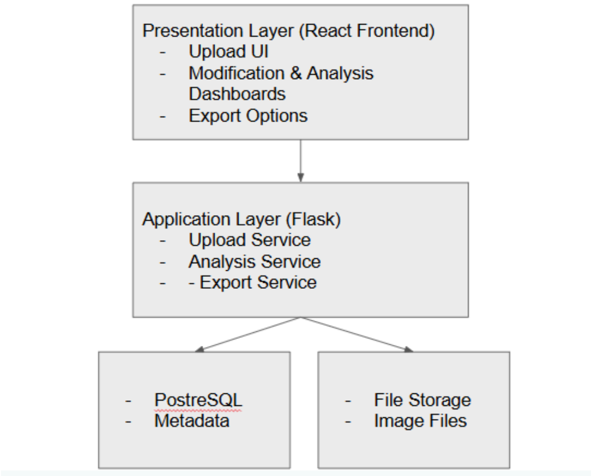


Figure 7: Software three-tier architecture diagram

3.3 Development Environment, Tools, Languages, and Libraries

Selected technologies:

- Frontend: React, JavaScript, Fabric.js for image alterations, Material-UI for a user-friendly interface
- Backend: Python (CNN compatible), Flask framework
- Machine Learning Integration: Pytorch
- Database: PostgreSQL
- Integrated Development Environment: Visual Studio Code

3.4 Initial Software Test Plan

Overview

This testing plan ensures the application meets quality standards and effectively serves researchers. The plan incorporates multiple testing methodologies throughout the development lifecycle.

1. Unit Testing

   Purpose: verify individual components and functions work correctly in isolation
   Significance: catches bugs early, ensures code reliability, and facilitates refactoring
   Timeline: continuous throughout development (Weeks 1-14)
   Frameworks: PyTest

2. Integration Testing

   Purpose: Verify components work together correctly and data flows properly between systems
   Significance: Ensures seamless user experiences and prevents system-level failures
   Timeline: Bi-weekly during development

3. User Acceptance Testing

   Purpose: Validate the application meets psychology researchers' actual needs and workflows
   Significance: Critical for ensuring product usefulness and adoption by target users
   Timeline: TBD

4 Project Risks

Risk 1: CNN Model Integration Complexity

1. The Convolutional Neural Network models must be successfully integrated into the application architecture. Failure to properly integrate these models could result in incorrect analysis outputs, system crashes, or inability to process images, which would render the core functionality of the application unusable.
2. Preventative Actions
   a. Conduct early prototype integration testing with the CNN models in the first month.
   b. Understand model specifications and expected outputs.
   c. Create comprehensive unit tests for all integration points.
3. Monitoring Approach
   a. Code reviews focusing on integration of code quality.
   b. Testing to validate CNN input/output formats.
4. Trigger Events
   a. Testing reveals incompatible data formats
   b. CNN processing times exceed the expected.
   c. Output formats don't match research requirements.
5. Mitigation Plan
   a. Create potential fallback processing with simplified analysis if full integration fails.

Risk 2: User Interface Usability for Non-Technical Users

1. The primary users (psychology students/faculty researchers) may lack computer science expertise. If the interface is not sufficiently intuitive, users may struggle to utilize the application effectively, leading to low adoption rates and failure to meet the project's core objective of providing a user-friendly tool.
2. Preventative Actions
   a. Follow established UI/UX design principles for research applications.
   b. Create interactive prototypes before implementing full functionality.
   c. Implement comprehensive documentation.
3. Monitoring Approach
   a. Possibly schedule usability testing sessions with subject matter experts.
   b. Collect user feedback after demos
4. Trigger Events
   a. Users take more than x amount of time to complete a basic image upload.
   b. Usability testing reveals confusion in core features by >50% of testing users.
5. Mitigation Plan

    a. Simplify the interface by reducing visible options.

Risk 3: Data Security and Privacy

1. The application will process research images that may contain sensitive experimental data. A data breach or loss could compromise research integrity, violate university data policies, and damage trust with the Psychology Department
2. Preventative Actions
   a. Follow Gonzaga University Data Security and Privacy Policies.
   b. Store all data with appropriate security and access.
3. Monitoring Approach
   a. Track compliance with university IT security policies
4. Trigger Events
   a. Users access images that were uploaded by other users.
   b. Unauthorized access detected.
5. Mitigation Plan
   a. Implement additional authentication layers (user log-in)

Risk 4: Scope Creep

1. The project sponsor or development process may introduce new feature requests that expand the project beyond its initial scope. This can lead to missed deadlines for core functionalities, reduced software quality, and team burnout.
2. Preventative Actions
   a. Maintain and strictly prioritize a clearly defined product backlog.
   b. Implement a formal change control process where all new requests are reviewed and approved by the team and sponsor before being added to the backlog.
   c. Clearly define and agree upon sprint goals with the sponsor prior to each development cycle.
3. Monitoring Approach
   a. Track the volume and frequency of changes to the product backlog.
   b. Monitor sprint velocity and completion rates against initial plans.
4. Trigger Events
   a. Unplanned features consistently prevent the completion of sprint goals.
   b. The team frequently requires overtime to deliver on initial commitments.
5. Mitigation Plan
   a. Document new requests in the backlog for future consideration but do not incorporate them into the current sprint.
   b. Communicate the impact of scope changes on the timeline and existing features to the sponsor, and collaboratively re-prioritize the backlog.

Risk 5: Team Dependency and Knowledge Silos

1. If critical components of the system are owned by a single team member, the project faces significant risk if that member becomes unavailable (due to illness, departure, or overload). This can halt progress on key features.
2. Preventative Actions

    a. Implement pair programming and mandatory cross-functional code reviews.
    b. Maintain clear, up-to-date technical documentation and code comments.
    c. Ensure all critical modules have a primary and secondary owner.
3. Monitoring Approach
    a. Review task distribution during stand-up meetings.
    b. Monitor version control commit history to ensure multiple contributors are familiar with key parts of the codebase.
4. Trigger Events
    a. A team member is unable to contribute for an extended period.
    b. A specific task is consistently blocked because only one person has the knowledge to complete it.
5. Mitigation Plan
    a. Immediately initiate knowledge transfer sessions to bring other team members up to speed on the critical component.
    b. Reassign tasks and adjust the project timeline if necessary.

5 Initial Product Release Plan

5.1 Major Milestones

We begin with foundational infrastructure (Milestone 1) establishing the desktop application framework and local file management system. CNN integration (Milestone 2) is prioritized early as it represents the highest technical risk and is critical for stakeholder feedback. User interface development (Milestone 3) builds on the working backend to enable meaningful user testing. Feature completion (Milestone 4) allows adequate time for usability testing with psychology researchers. Final testing and distribution packaging (Milestones 5-6) ensure quality assurance and smooth handoff to the Psychology Department.

Table 3: Major Milestones

Milestone	*Description*	*Target Completion Date*
Foundation Infrastructure	Establish desktop application framework, local file management system, user settings/preferences storage, and basic application navigation. Create a development environment with cross-platform build tools. Deliverable: Working desktop prototype with basic UI and file browsing.	First week of December
CNN Integration and Basic Analysis	Successfully integrate existing Psychology Department CNN models into desktop applications. Implement a local image processing pipeline with similarity analysis and results generation. Deliverable: Functional desktop application that analyzes images and generates similarity scores stored locally.	Third week of January

User Interface and Visualization	Complete desktop UI with all primary screens including main dashboard, image library, and results viewer. Implement similarity matrix visualization and heatmap generation with local image display. Deliverable: End-to-end workflow from local image selection through result visualization in desktop application.	Second week of February
Feature Completion and Enhancement	Implement remaining major features including image editing tools, CSV export, project folder management, and advanced complexity metrics. Complete integrated help documentation. Deliverable: Feature-complete desktop application ready for comprehensive testing.	Third week of March
User Testing and Refinement	Conduct comprehensive usability testing with psychology students and faculty on their own computers. Iterate on design based on feedback. Perform performance testing on different hardware configurations. Address all critical bugs and usability issues. Deliverable: Tested, refined desktop application validated by target users on Windows and macOS.	Second week of April
Packaging and Distribution	Create installation packages for Windows (.exe installer) and macOS (.dmg). Develop installation guide and troubleshooting documentation. Conduct installation testing on various computer configurations. Final acceptance testing with Dr. Addleman. Deliverable: Distributable desktop application with complete installation documentation and trained users.	First week of May

5.2 Initial Sprint Releases

Table 4: Sprint Release Plan

Sprint Date	*Spring Goal*	*Backlog*	*What we will demo*
4th Week in Oct to 1st week in Nov	Establish Foundation and Development Environment	Desktop Application Framework (partial), Development environment setup, Database schema design	Basic desktop window application with menu bar, file browser for selecting images, and local storage of user preferences. Demo on team member laptops and gather feedback on overall desktop UI layout.
2nd Week in Nov to 3rd Week in Nov	Local Image Management System	Image Upload/Import System, Basic Image Display, Local project folder management	Desktop application that can browse and import images from local file system, display thumbnails in organized view, and save project state locally. Demo importing psychology research images from USB drive or local folders.

4th Week in Nov to 1st Week in Dec	CNN Integration Breakthrough	CNN Model Integration (primary focus), Local processing pipeline	Functional desktop application that processes local images through integrated CNN models and displays similarity scores. Demonstrate processing speeds on laptop hardware. Address Risk 1 (CNN Integration Complexity) through working prototype. **Milestone 1 Complete**
2nd Week in Dec to 3rd Week in Dec	Results Display and Semester Wrap	Similarity Analysis Results (partial), Results visualization UI	Desktop application showing similarity matrix in dedicated results window, with ability to save results locally as CSV or image files. Gather feedback on result interpretation clarity before winter break.
2nd Week in Jan to 3rd Week in Jan	Visualization Enhancement	Similarity Analysis Results (complete), Heatmap Visualization	Complete results viewer with interactive similarity matrix, heatmap overlays displayed over images in desktop viewer. Demo various heatmap color schemes and export options. **Milestone 2 Complete**
4th Week in Jan to 1st Week in Feb	Export and Organization Features	CSV Export Functionality, Project Organization Features, File management	Desktop application with robust CSV export, project save/load functionality using local file system, and organized workspace for managing multiple research studies. Demo saving and reopening projects.
2nd Week in Feb to 3rd Week in Feb	Image Editing and Advanced Features	Basic Image Editing Tools, Advanced Complexity Metrics	Built-in image editor with brightness/contrast/crop tools saving modified images locally, extended complexity analysis metrics. Demo creating experimental stimulus variations within the application. **Milestone 3 Complete**
4th Week in Feb to 1st Week in Mar	Documentation and Polish	User Documentation System, UI refinements, Context-sensitive help	Integrated help system accessible from desktop menu bar, polished desktop interface with improved keyboard shortcuts and workflows. Demo complete user journey including F1 help access. Address Risk 2 (User Interface Usability).
2nd Week in Mar to 3rd Week in Mar	Usability Testing Sprint	Conduct Usability Testing on user machines, Bug fixes, Cross-platform refinements	Install application on psychology students' computers (3-5 participants) for real-world testing, collect feedback on different

			hardware/OS configurations, demonstrate implemented improvements. Discuss testing findings with sponsor. **Milestone 4 Complete**
4th Week in Mar to 1st Week in Apr	Performance and Installation Testing	Performance optimization for various hardware, Installation package creation (alpha), Cross-platform testing	Test application on different computer specifications (older laptops, newer desktops), create initial Windows .exe and macOS .dmg installers, demonstrate installation process. Address file access permissions and security.
2nd Week in Apr to 3rd Week in Apr	Packaging and Distribution Prep	Final installation packages, User Manual completion, Installation guide	Completed installers for Windows and macOS with proper code signing, comprehensive user manual (PDF), step-by-step installation guide with screenshots. Conduct test installations on clean machines. **Milestone 5 Complete**
4th Week in Apr to 1st Week in May	Final Testing and Handoff	Final acceptance testing, Developer documentation, Training sessions	Distribute application to initial user group, conduct in-person training session with psychology department, provide all source code and build instructions. Final acceptance and project handoff. **Milestone 6 Complete**

6 Maintenance Considerations

1. Software Maintenance (Technical Updates)

Maintaining the desktop application and CNN models as technology evolves. This includes:

- Application Updates: Security patches, bug fixes, and OS compatibility updates for new Windows/macOS versions
- CNN Model Updates: Retraining models with new datasets or adding new models for emerging research questions
- Dependency Management: Updating Python libraries and frameworks while maintaining compatibility
- Performance Optimization: Improving processing speed and memory usage based on user feedback

Required Expertise: Desktop application development (Python, GUI frameworks), basic machine learning knowledge, and cross-platform build tools. Estimated skill level: Junior/Senior CS student with desktop development coursework.

2. User Support and Training

Supporting psychology researchers in using the application effectively:

- Installation Assistance: Helping new students install the application on their computers each semester
- Troubleshooting: Resolving technical issues on different computer configurations
- Training: Conducting training sessions for new users at the start of each semester
- Documentation Updates: Creating additional guides and FAQs based on common user questions

Required Expertise: Familiarity with the application features, good communication skills for non-technical users, basic troubleshooting abilities. Estimated skill level: Psychology graduate student or CS student with technical writing skills.

7 Project Management Considerations

Our team will follow an agile sprint setup to stay organized and ensure consistent progress toward project deliverables. We will hold weekly team meetings with our faculty advisor, meet at least once a month with our sponsor, contact our DAB advisor as needed, and hold student stand-up meetings throughout the week to plan sprints, review completed work, and assign new tasks. Meetings will happen over Zoom or on campus in classrooms or reserved rooms (Usually Monday 5:30pm). Communication between student team members will occur primarily through iMessage for quick coordination. We will give regular updates to advisors and sponsors via email. GitHub Projects will be used to manage our product backlog and sprint backlog, allowing us to track tasks, progress, and milestones transparently. Each team member will take ownership of specific components of the system (a specific piece of the frontend or backend), ensuring accountability and balanced workload distribution. A team member will then have their code reviewed by at least one other member before integrating their work into the Git repository. In addition to GitHub, we will use shared documents and notes to communicate progress and collect feedback from our sponsor, advisors, and other stakeholders throughout development.

Team Member Bios

Leonard Jia

Major: Computer Science

Hometown: Bellevue, WA

Education:

High School: Liberty High School

University: Gonzaga University, Senior

Applicable Courses:

Algorithms and Abstract Data Structures, Software Development, Database Management Systems, Applied Statistical Models, Data Science Algorithms, Data Science Project Lab, Organization of Programming Languages

Work Experience:

Fast Enterprises | Implementation Intern | Summer 2025

Developed C# code for the Idaho Division of Occupational and Professional Licensing, supporting over 300,000 license accounts. Wrote optimized SQL queries to extract, validate, and transform data, improving system performance and accuracy. Consulted with 48 boards and commissions to gather requirements and ensure alignment with proper business processes.

Skills:

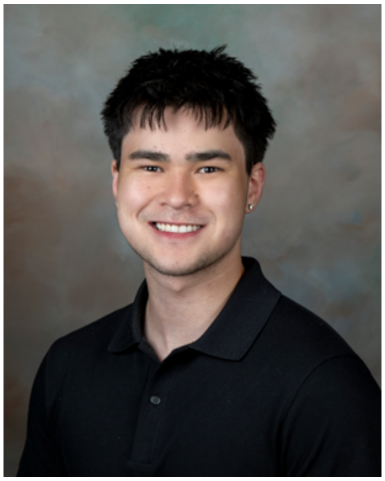
Programming Languages: C#, C++, Java, Python, HTML, CSS, SQL, R, NodeJS

Development Tools: Github, VSCode, Docker, PGAdmin, RStudio, Firebase

Other Interests:

Hackathon Participation: 2 year hacker for Stanford University TreeHacks with projects including Fitstreak – AI-powered fitness tracking, Culinary Connections – AI-powered recipe sharing and recommendations.

Hobbies: Pokémon TCG, Video Games, Poker, Electronic music



Jacob Jarvis

Major: Computer Science

Minor: Applied Math

Hometown: Newcastle, WA

Education:

High School: Liberty High School

University: Gonzaga University, Senior

Applicable Courses:

Algorithms and Data Structures, Software Development, Database Management System, UI/UX, Computer Security, Linear Algebra, Calculus

Work Experience:

NASA Armstrong Research Center (June 2025 – August 2025)

Computer Science Research Intern - Developed a dual-phase computer vision system for NASA's AIRVUE platform using YOLO object detection and U-Net semantic segmentation models. Engineered a comprehensive data pipeline integrating 7 heterogeneous datasets and increased image segmentation model accuracy by ~50% through iterative optimization.

Skills:

Programming Languages - C++, Java, Python, C#, React, Flask, JavaScript, HTML, CSS, SQL, XML

AI/Machine Learning - PyTorch, TensorFlow, YOLO, U-Net, Google Colab, Computer Vision

Development Tools - Github, VS Code, Google Cloud Storage, Database Management

Other Interests:

Hackathon Participation - Active participant in TreeHacks hackathon with projects including Cloak AI (private LLM conversations using vector databases) and FitStreak (AI-powered fitness tracking platform)

Hobbies – Reading, Video Games, Weight-lifting, Formula 1, Pokémon, Technology

Zobe Murray

Major: Computer Science

Hometown: Bothell, WA

Education:

High School: Bothell High School

University: Gonzaga University Senior

Applicable Courses: Data Science Algorithms, Software Development, Machine Learning and Intelligence Systems

Work Experience:

Knight Cancer Research Institute – OHSU (June-August 2024)

Research Intern - Built a microbiome analysis pipeline to assess the mycobiome in patients with head and neck cancer, enabling detailed data processing and visualization.

Biology Department – GU (January – September 2024)

Research Assistant – Collaborated with Professor Christy Andrade on a next-generation sequencing (NGS) project to analyze microbial communities in mosquito larvae and their aquatic environments.

Institute for Informatics and Applied Technology – GU (June-August 2025)

Research Assistant – Collaborated with Professor Doug Addleman to implement VGG16 and ResNet50 convolutional neural network models to calculate and analyze image similarities.

Skills:

C++, Java, Python, R, SQL, Some HTML – Languages that I have learned throughout my academics and work experience

Google Colab, VS Code, IntelliJ, R Studio, Git (GitHub), Docker– Systems/Applications I have experience using

Interests:

- Data Analytics and Contributing to interdisciplinary research

Junyou Guo
Major: Computer Science
Hometown: Hangzhou, China

Education:

High School: Zhejiang Hangzhou High School

University: Gonzaga University, Senior

Applicable Courses:

Algorithms and Abstract Data Structures, Software Development, Database Management Systems, Data Science Algorithms, Mobile App Development

Work Experience:

Tencent CSIG | Backend Development Intern | Summer 2024

Drove the adoption of shared code across a 200-member department, improving software rollout speed by ~30% and reducing redundant work.
Leveraged knowledge of data structures and project management to streamline workflows, enhancing team productivity and cross-team collaboration efficiency.

Skills:

Programming Languages: C++, Java, Kotlin

Development Tools: Github, VSCode, Android Studio

Interests:

Club: one of the establishers of GonzagaU Esports Club

Hobbies: video games, outdoors