

# Problem Statement and Goals

## ProgName

Team #, Team Name  
Student 1 name  
Student 2 name  
Student 3 name  
Student 4 name

Table 1: Revision History

Date	Developer(s)	Change
09/13/2025	Omar El Aref(s)	Added the problem statement, Inputs and Outputs
09/14/2025	Omar El Aref(s)	Added Stakeholders, Environment, Goals, Stretch Goals and Extras
09/22/2025	Omar El Aref(s)	Added Reflection
...	...	...

## 1 Problem Statement

[You should check your problem statement with the [problem statement checklist](#). —SS]

[You can change the section headings, as long as you include the required information. —SS]

### 1.1 Problem

The textual history of Indian Buddhism is fragmented across thousands of manuscript folios, preserved only in partial, damaged, or scattered forms. Traditionally, scholars reconstruct these texts manually through palaeographic study, transcription, and content comparison; a slow and error-prone process. While some existing computational tools exist for pattern recognition and Optical Pattern Recognition (OCR) none can tackle this problem as they are not designed for irregular, damaged, or arbitrarily oriented manuscript fragments.

The lack of computational tools available for this problem significantly limits progress in reconstructing Buddhist textual history. Scholars need a tool that

automates the detection, matching, and transcription of manuscript fragments, thereby reducing manual effort and time; enabling large-scale reconstruction.

## 1.2 Inputs and Outputs

- **Inputs:**

- High quality images of manuscript fragments (approximately 21,000 images from collections)
- Metadata (collection identifiers, orientation, partial transcriptions if available)

- **Outputs:**

- Probabilistic matches between fragments based on shape/edge/damage features (i.e., list of fragments most likely related to an input fragment image)
- Enhanced Metadata:
  - \* Suggested transcriptions of fragment text
- Searchable/sortable database containing fragment attributes and relationships

[Characterize the problem in terms of “high level” inputs and outputs. Use abstraction so that you can avoid details. —SS]

## 1.3 Stakeholders

- **Primary Stakeholders:**

- Scholars of Buddhist textual history (religious studies, philology, palaeography)
- Supervisors and domain experts (e.g., Dr. Shayne Clarke, McMaster Religious Studies)

- **Secondary Stakeholders:**

- Computer science researchers in AI, image processing
- Humanities researchers studying textual transmission and manuscript culture

- **End Users:**

- Academic researchers using the software to assemble fragments
- Graduate students seeking computational support in philological research

## 1.4 Environment

- **Hardware:**
  - Development laptops/workstations with GPU support for ML tasks
- **Software:**
  - VScode
  - Coding Libraries
  - Database
  - GitHub repository for version control and CI/CD

[Hardware and Software Environment —SS]

## 2 Goals

- Develop a tool that:
  - Detects edges, shapes, and damage patterns in fragments.
  - Matches fragments based on similarity measures (probabilistic scoring).
  - Identifies palaeographic script of fragments.
  - Performs preliminary transcription using OCR tuned to Sanskrit scripts.
  - Builds a searchable database linking fragments with metadata and probable matches.
  - Provides a user interface for scholars to view suggested fragment matches and confirm/annotate them.

## 3 Stretch Goals

- Expand support to Tibetan and Chinese manuscript fragments.
- Incorporate semantic content matching with.
- Improve transcription accuracy with AI-assisted error correction.

## 4 Extras

- **User Documentation:** Write user-friendly guides for scholars with limited technical expertise.
- **Use Case Video:** Make a Video on how to use the tool so that we can minimize the learning curve for the intended user

[For CAS 741: State whether the project is a research project. This designation, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[For SE Capstone: List your extras. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. (The full list is on the course outline and in Lecture 02.) Normally the number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

## Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
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)?