Development Plan
Com-puter Vision

Team # 19, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

September 18, 2025

Table 1: Revision History

Date	**Developer(s)**	**Change**
2025-09-18	All	Initial draft of Development Plan
...	...	...

1 Development Plan Overview

The development plan for Com-puter Vision outlines how our team will organize, develop, nd deliver an AI-based system to identify cows by their black-white patterns on their back and detect potential health issues. This document includes our Confidential information, IP to protect, Copyright license, Team meeting plan, Team communication plan, Team member roles, Workflow plan, Project decomposition and scheduling, Proof of concept demonstration plan, Expected technology, and Coding standard.

[Additional information on the development plan can be found in the lecture slides. —SS]

2 Confidential Information?

Our project does not involve confidential information from industry. All data and models will be based on publicly available datasets or collected in compliance with open-source or academic research guidelines.

3 IP to Protect

There is no proprietary intellectual property (IP) to protect for this project. The outputs will be published under an open-source license to encourage collaboration and reproducibility.

4 Copyright License

The team has chosen the MIT License, which will be included in the GitHub repository. This license provides flexibility for others to use, modify, and distribute the project with proper attribution.

5 Team Meeting Plan

- Team check-in meetings will be held twice weekly via Microsoft Teams or Zoom, all team members are expected to attend and share their progress.
- An in-person meeting will be scheduled at least once per month on campus.
- The industry advisor will meet with the team weekly, virtually. Including check-in and Q&A.
- Each meeting will have a designated chair and note-taker, with an agenda shared in advance.

6 Team Communication Plan

- Primary communication will be through MS Teams for quick messages and updates.
- Email will be used for formal communications and sharing important documents.
- Regular progress will be shared in a dedicated GitHub repository.
- GitHub Issues will be used to track lecture attendance, tasks, bugs, and feature requests.
- A shared file folder on MS Teams will be used for collaborative document editing and storage.

7 Team Member Roles

Every team member are responsible for attending meetings, replying to messages, creating issues, coding, testing, reviewing code, and documentation. In addition, team members will take on the following rotating roles:

Meeting Chair

Leads meetings and ensures meetings stay on track. Every team member will take turns being the chair.

Note-Taker

Records meeting minutes, captures action items, and distributes notes to the team. Every team member will take turns being the note-taker.

Reviewer

Reviews code contributions, provides feedback, and ensures coding standards are met. Zongcheng Li, Ji Zhang and Baoning Zhang will primarily take on this role, but all team members are encouraged to participate in code reviews.

Integrator

Merges code contributions, resolves conflicts, and ensures the codebase remains stable. Changhao Wu and Zongcheng Li will primarily take on this role, but all team members are

Lead Developer

Oversees the technical direction of the AI/ML pipeline, coordinates integration between modules, and supports other members in debugging and implementation. Changhao Wu and Zongcheng Li will primarily take on this role, but all team members are encouraged to contribute to technical decisions. All team members will contribute to development tasks.

Communication Lead

Prepare agendas, manages communication with the industry advisor, schedules meetings, and ensures Zongcheng Li and Changhao Wu will primarily take on this role, but all team members are encouraged to participate in communication tasks.

8 Workflow Plan

The following workflow will be adopted for the project: 1. Pull any changes from the main branch to ensure the local repository is up-to-date. 2. Create a new branch for each feature or bug fix, following the naming convention 'feature/feature-name' or 'bugfix/bug-name'. 3. Create a draft of the structure of the code/documentation. 4. Implement the functions without dependencies first, then the functions with dependencies. 5. Commit changes frequently with clear, descriptive messages. 6. Push the branch to the remote repository on GitHub. 7. Open a pull request (PR) to merge the feature/bugfix branch into the main branch. 8. Assign at least one team member as a reviewer for the PR. 9. Address any feedback from the reviewer(s) and make necessary changes. 10. Once approved, the PR will be merged into the main branch. 11. Delete the feature/bugfix branch after merging to keep the repository clean. 12. Use GitHub Issues to track tasks, bugs, and feature requests, with appropriate labels for classification. 13. Update any changes in the weekly check-in meetings.

9 Project Decomposition and Scheduling

- How will you be using GitHub projects?
- Include a link to your GitHub project

[How will the project be scheduled? This is the big picture schedule, not details. You will need to reproduce information that is in the course outline for deadlines. —SS]

10 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

11 Expected Technology

[What programming language or languages do you expect to use? What external libraries? What frameworks? What technologies. Are there major components of the implementation that you expect you will implement, despite the existence of libraries that provide the required functionality. For projects with machine learning, will you use pre-trained models, or be training your own model? —SS]

[The implementation decisions can, and likely will, change over the course of the project. The initial documentation should be written in an abstract way; it should be agnostic of the implementation choices, unless the implementation choices are project constraints. However, recording our initial thoughts on implementation helps understand the challenge level and feasibility of a project. It may also help with early identification of areas where project members will need to augment their training. —SS]

Topics to discuss include the following:

- Specific programming language
- Specific libraries
- Pre-trained models
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Tools you will likely be using?

[git, GitHub and GitHub projects should be part of your technology. —SS]

12 Coding Standard

[What coding standard will you adopt? —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. Why is it important to create a development plan prior to starting the project?
2. In your opinion, what are the advantages and disadvantages of using CI/CD?
3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

[What are your team's external goals for this project? These are not the goals related to the functionality or quality fo the project. These are the goals on what the team wishes to achieve with the project. Potential goals are to win a prize at the Capstone EXPO, or to have something to talk about in interviews, or to get an A+, etc. —SS]

Attendance

Expectations

[What are your team's expectations regarding meeting attendance (being on time, leaving early, missing meetings, etc.)? —SS]

Acceptable Excuse

[What constitutes an acceptable excuse for missing a meeting or a deadline? What types of excuses will not be considered acceptable? —SS]

In Case of Emergency

[What process will team members follow if they have an emergency and cannot attend a team meeting or complete their individual work promised for a team deliverable? —SS]

Accountability and Teamwork

Quality

[What are your team's expectations regarding the quality of team members' preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team's expectations regarding team members' ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]