Introduction

Team Project Overview

Team 34 is developing an autonomous book reshelving system for library environments. The system integrates vision-based perception, task planning, and robotic manipulation to automate the return process. Unlike traditional systems that rely on fixed coordinates, our system perceives the shelf in real time, detects available gaps, and makes placement decisions based on current conditions. The workflow includes spine-based book identification, RGB-D shelf perception, intelligent slot selection, and visual-servo placement control. After a single user input, the system processes a batch of returned books without manual intervention.

Individual Responsibilities and Role

My role in Team 34 is Perception & Visual-Servo Lead. I am responsible for the entire software pipeline from camera frame to 3-D target coordinate: the Book Identification module, the Shelf Occupancy Perception module, the Slot Selection Policy, and the Position-Based Visual Servoing controller. I also maintain the perception-side adapter that connects my pipeline to the main program Zhenxiong maintains. Over the past two weeks my primary effort has been on the Book Identification module, specifically a PaddleOCR-based spine text recognition pipeline with character-level localization.

Individual Design Work

Book Identification Pipeline

I finalized the book identification pipeline using PaddleOCR as the single recognition model. The Design Document's Section 2.2.1 originally specified a two-stage YOLOv8 + PaddleOCR cascade. After benchmarking PaddleOCR's built-in DBNet detector and CRNN recognizer, I dropped the YOLOv8 stage: PaddleOCR's own detector handles mixed-orientation text lines natively, and removing the extra stage cut a noticeable amount of per-frame CPU inference time with no accuracy loss. The recognition target was also broadened from fixed classification labels (e.g., "A1", "B3") to actual Chinese and English book titles, which generalizes beyond a closed label set; color matching is retained only as a fallback.

The pipeline operates on a fixed vertical ROI centered in the RGB frame. Because book spines stand upright, this ROI matches spine geometry, crops inference to a small fraction of the full frame, and suppresses background clutter from neighboring books and shelf edges. A green overlay on the live preview indicates where the arm should position the spine before capture.

Adaptive Character-Level Localization

PaddleOCR returns one polygon per text line, but I need per-character boxes so that the BookSync database query can tolerate single-character misreads through fuzzy substring matching, and so that OCR failures can be diagnosed visually. Real spines mix vertical Chinese titles with horizontal English labels, so a hard-coded text axis would fail on half of every book. I wrote an adaptive character-box function that determines each line's long axis at runtime by comparing the two adjacent edge lengths of the line polygon, then interpolates character corner points along the detected axis. The same function handles both orientations without an explicit language flag.

Testing and Verification Plans

Current testing focuses on the OCR module. Planned tests include OCR accuracy over a batch of books under controlled indoor lighting, targeting the identification accuracy specified in the Design Document; orientation robustness over books presented at multiple rotations; end-to-end OCR latency measured across repeated captures; and database query latency for the BookSync lookup. Preliminary functional validation confirmed the pipeline runs end-to-end on sample spines; quantitative accuracy and latency measurements are scheduled for Week 5. The per-character decomposition is designed to recover from single-character misreads through fuzzy substring matching against the database. Contingency plans include fine-tuning PaddleOCR's recognition head on a small dataset of our test books, adding a BookSync category prior to constrain candidates, and adding a USB LED ring light if lighting becomes the limiting factor.

Conclusion

Self-Assessment

I estimate my contribution at approximately 30% of the total project workload, concentrated almost entirely on the vision software stack — effectively all of the project's computer-vision content. Current progress is slightly behind the Design Document schedule, with the slip localized to OCR accuracy tuning; the rest of my scope (shelf occupancy, slot selection, PBVS) remains on schedule. The delay is not on the integration critical path, because the perception-adapter contract only requires a valid book identifier and target shelf from identification; OCR tuning can continue in parallel with the upcoming shelf-perception work.

Plans for Remaining Work

Current focus is closing the OCR accuracy, orientation, and latency tests and implementing the column-wise depth gap detector, so that the adapter can begin emitting structured output to the decision subsystem. The next stage is completing the density-maximization slot selector and running the first integration test with Zhenxiong over synthetic shelf states. The final stage is Tsai–Lenz hand-eye calibration followed by PBVS convergence testing, targeting the placement accuracy specified in the Design Document, and then the end-to-end batch run within the time and success-rate targets. Hand-eye calibration is the highest-risk item, so I am pre-allocating a buffer by advancing the emergency-stop vision-shutdown subtask earlier in the schedule.