# 📝 NLP Project – Technical Lead Report (Week 3-4)

**Name**: Erin David Cullen

**Start Date**: 04 August

**Date**: 17 August

## ✅ 1. Work Completed (Since Last Report)

\_What you’ve built, implemented, tested or reviewed and an estimate as to how many hours it took to complete

* System architecture refactoring and Flask migration (~16-20 hr)
* WebUI implementation (~2 hr)
* Performance optimization - YOLOv11 migration and threading (~10-12 hr)
* Depth detection integration with Structure From Motion (~10 hr)
* **Feature/Module:** System Architecture Refactoring
  + Migrated from monolithic AI.py to modular Flask-based architecture
  + Implemented threaded main method with separate video, audio, and processing pipelines
  + Created comprehensive system flowchart documenting all components and data flow
  + Established global state management for inter-thread communication
* **Feature/Module:** Web Control Interface
  + Built Flask-based web control system with real-time monitoring capabilities
  + Implemented WebUI for drone control and system status visualization
  + Created responsive control interface accessible via web browser
  + Integrated live video feed display with AI overlay rendering
* **Feature/Module:** Performance Optimization - YOLOv11 Migration
  + Migrated object detection system from YOLOv8 to YOLOv11 for improved accuracy and performance
  + Optimized object detection to process frames at 100ms intervals (10 FPS processing)
  + Maintained full FPS live video stream with periodic AI analysis overlay
  + Implemented efficient frame extraction and processing queue system
  + Enhanced threading architecture for improved real-time performance
* **Feature/Module:** Depth Detection & Structure From Motion
  + Integrated depth analysis that accumulates object detection results over time
  + Implemented Structure From Motion algorithms for distance measurement
  + Created depth comparison utilities for object distance calculation
  + Established depth node integration with global state management system
* **Tools/Libs Used:**
  + Flask framework for web application architecture
  + Threading libraries for concurrent processing pipelines
  + OpenCV for video processing and Structure From Motion
  + YOLOv11 (ultralytics) for enhanced object detection performance
  + WebSocket for real-time web interface communication
  + Global state variables for inter-thread data sharing
* Outcome/Results:
  + Successfully refactored system into modular, maintainable Flask architecture
  + Achieved real-time object detection with 100ms processing intervals while maintaining smooth video playback
  + Implemented working depth detection using Structure From Motion techniques
  + Created comprehensive web-based control interface for system monitoring and control
  + Established robust threading architecture supporting concurrent video, audio, and AI processing
* Contributions (if applicable):
  + Designed and implemented complete system architecture refactor
  + Built Flask-based web control interface with real-time capabilities
  + Optimized object detection pipeline for improved performance
  + Integrated depth detection with object tracking accumulation
  + Created comprehensive system documentation and flowchart
  + Established threading architecture for concurrent processing

## 📌 2. Current Tasks in Progress

\_What you’re actively working on. Include blockers if any.\_

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Description | ETA | Blockers |
| YOLOv11 fine-tuning | Optimizing YOLOv11 model parameters for drone-specific detection scenarios | ~4 hours | Model compatibility testing |
| Depth accuracy validation | Testing Structure From Motion accuracy across different scenarios | ~6 hours | Need diverse test environments, camera stabilization for smooth motion |
| Threading optimization | Fine-tuning concurrent processing efficiency and synchronization | ~4 hours | Thread synchronization complexity |

## 📅 3. Upcoming Tasks

\_Planned work for the next sprint or phase.\_

* **Task:** Command Processing Integration
  + **Purpose/Goal:**
  + Integrate voice command processing with depth detection pipeline
  + Implement command status flow management ("new" → "processing" → "complete")
  + **Dependencies:**
  + Stable depth detection system
  + Refined global state management
* **Task:** Real-time Performance Optimization
  + **Purpose/Goal:**
  + Optimize threading performance to maintain consistent 100ms processing intervals
  + Implement advanced frame buffering and queue management
  + **Dependencies:**
  + Current threading architecture stability
  + Performance profiling and bottleneck identification
* **Task:** SLAM Integration Enhancement (Optional)
  + **Purpose/Goal:**
  + Fully integrate RTAB-Map SLAM with new Flask architecture
  + Ensure SLAM runs continuously in background thread
  + **Dependencies:**
  + Threading architecture finalization
  + SLAM node compatibility with new system structure

## 🚨 4. Issues & Risks

\_Bugs, technical debt, resourcing, or anything threatening progress.\_

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Impact | Suggested Action | Owner |
| Video processing latency | Delays between live video and AI processing overlays affect real-time performance | Optimize frame buffering and reduce processing overhead | Ed |
| Inconsistent FPS performance | Variable frame rates affect smooth video playback and user experience | Implement adaptive frame rate control and resource management | Team |
| Video quality degradation | Processing pipeline reduces video quality affecting detection accuracy | Optimize compression and maintain higher quality throughout pipeline | Team |
| Inaccurate depth detection | Structure From Motion calculations significantly off from actual distances | Recalibrate depth algorithms and improve feature matching accuracy | Ed |

## 📈 5. Key Insights / Recommendations

\_Lessons learned, suggestions, architecture notes, or optimizations.\_

* Flask-based architecture provides significantly better modularity and maintainability compared to monolithic approach
* Threading architecture with global state management enables efficient concurrent processing of video, audio, and AI pipelines
* 100ms object detection intervals provide optimal balance between real-time responsiveness and system performance
* Structure From Motion depth detection works effectively when combined with object tracking accumulation over multiple frames
* Web-based control interface greatly improves system usability and real-time monitoring capabilities