

AstrID – Week 5 Plan - Meeting 5

What I Built This Week

Got the standalone training pipeline working! Now I can train U-Net models for anomaly detection without needing the full AstrID system running.

Key Stuff Delivered

- **Synthetic data generator** - Creates realistic astronomical images with transients, variable stars, supernovae, asteroids
- **U-Net model** - Full encoder-decoder architecture with ~2.1M parameters
- **Training pipeline** - Complete with MLflow tracking, early stopping, model saving
- **Real data integration** - Can fetch actual images from MAST/SkyView APIs
- **Evaluation system** - All the metrics (accuracy, precision, recall, F1, AUC, etc.)

How to Use It

The system has different modes - there's a quick test mode that runs with just 20 samples and 3 epochs to verify everything works, a full training mode for more serious runs, and a custom mode where I can specify exactly how many samples and epochs I want. It's pretty straightforward to use.

This Week's Plan

1. **Test everything** - Run the test suites to make sure it all works
2. **Train some models** - Try different configurations to see what works best
3. **Analyze results** - Look at the metrics and see how well it's performing

Testing Commands

I have several test scripts set up - one for testing the basic pipeline, another for testing the MAST API integration for real data fetching, a demo script that shows off the capabilities, and the quick training test. All of these help me verify that everything is working correctly before doing serious training runs.

What's Working

- ☒ Standalone training (no external dependencies)
- ☒ Synthetic data generation with realistic features
- ☒ U-Net model for anomaly detection
- ☒ MLflow experiment tracking
- ☒ Real data from MAST API
- ☒ Comprehensive testing

Questions for Advisor

- **Architecture**: Does the U-Net design make sense for astronomical data?
- **Metrics**: Are these the right evaluation metrics for transient detection?

- **Performance:** What should I expect for precision/recall on real data?
- **Anomaly types:** Are the synthetic anomaly types I'm generating realistic enough?

Next Steps

- Get real data working with MAST API
- Compare synthetic vs real data performance
- Plan integration with main AstrID system

This gives me a solid foundation to work with - I can now develop and test models independently before integrating with the full system.